App FW Scope and Concept - Review

Wiki page for logging review comments and discussions.

Base version:

Concept document
GENIVI Application Framework

7th July 2016
Document Author: Simon McVittie
Document Reviewer: Sjoerd Simons
Document Reviewer: Philip Withnall
Document Reviewer: Guy Lunardi
Document Owner: Collabora - Guy Lunardi
E-mail: guy.lunardi@collabora.com
Tel: : \+1-801-200-3848
Collabora Document Number: GEN0001
Collabora Document Version: v0.3 (draft)
Author Date Version Change Approved
SM 29-06-2016 0.1 Initial version SM
SM 04-07-2016 0.2 Revised internal version SS
GL 23-06-2016 0.3 First draft shared with GENIVI GL
link to the document : Application Framework scope and requirements

---

Application Framework scope and requirements

This document outlines requirements relevant to the GENIVI Application Framework effort. However some of these requirements may well be considered out of scope for requirements to the GENIVI Application Framework due to overlap with other GENIVI initiatives. They are included here as they are perceived to be within the context of an application framework. This document does not aim to specify a particular implementation for any requirement. The terms privilege, privilege boundary, confidentiality, integrity and availability have their usual information-security meanings (for definitions, please refer to Apertis Security design). This document is authored by Collabora.

Table of Contents

What's in an app

NOTE: Initial IDs are assigned equal to the line number in the original document.

ID	Text
25	There are two commonly-used definitions of an "app": either a user-facing launchable program (an entry point) such as would appear in launcher menus, or a user-installable package or bundle such as would appear in an app store.
	Comments: GN - info - Within GENIVI scope, 'Managed Apps' and 'Native Applications' has been defined GA: A fair assumption is that a bundle will never contain both managed and native apps and an app is either native or managed. I think we are fine with the current definitions and they can be used in combination with the defintions here without any conflict between them. GA: A clear definition of "bundle" would help here (because of later usage e.g. entry-points) GM: Please clarify how this definition is related to "Managed Applications" and "Native Applications" as defined inside the GENIVI Reference Architecture GM: Also "package or bundle" should be better defined (notice that the GENIVI Compliance Specification already defines a "Package Manager") PW: 'bundle' is defined on line 27 as a concept similar to an app on Android, for example. The definition is refined over the course of the document. We could try and expand the summary here a bit. GA: This week 2016-11-08 it seems we said that Bundle = A collection of zero or more executable files, zero or more libraries and zero or more meta-data files (for the last case imagine for example a "Skinning/theming bundle containing Icon graphics and configuration files only.". In an app store what would be presented to the user as one app would technically be downloaded as one bundle of files.
	Extracted requirement:

ID	Text
28	A user-installable bundle would most commonly have exactly one entry point. However, it might not have any entry points at all, for example if it is a theme or some other extension for the operating system. Conversely, it might have more than one entry point: for example, a user-installable bundle for an audio player might contain separate menu items for music and audiobooks, launching different executables or the same executable in different modes to provide an appropriate UX for each use-case.
	Comments: GN : Ok, Multi entry points for Apps need to be translated into requirement. GA: Put definitions at top, e.g. what is a bundle according to our previous discussions.
	Extracted requirement:

ID	Text
34	In this document, when we need to distinguish between the two meanings, we will say that a user-installable bundle contains zero or more entry points. Entry points are similar in scope to Android activities.
	Comments: GN :OK GA: I am still not sure that the Android comment helps. From android docs: "An Android Activity is an application component that provides a screen with which users can interact in order to do something". Activity requires a graphical view? Is that also required for each entry point? Let's put down an actual definition table at the start of this document - for bundles, entry-points, etc. PW: You're right. Entry points do not require a graphical view. I agree with putting a definition of entry points at the start of the document.
	Extracted requirement:

ID	Text
37	Some vendors might decide to restrict the apps available in their app stores to have at most one entry point, but that is a policy decision by those vendors and should not be reflected in the more general app framework.
	Comments: GN : Should vendor specific config be a part of manifest? PW: Vendor-specific configuration should be a part of the vendor's platform policy, which will be encoded in their app store guidelines, and implementations of vendor-specific components in the platform. Manifests contain metadata, which should be kept separate from policy. It would be redundant to encode a general vendor-wide policy of "maximum number of entry points" in the manifest for each app. Type: INFO.
	Extracted requirement:

ID	Text
40	Entry points might be written as native code (for example compiled from C or C++), or they might run under an interpreter or JIT in a runtime environment that provides GUI functionality analogous to native code (for example if the app is written in Java, Python, or JavaScript for the node.js, gjs or seed runtime environments), or they might run in a HTML5 runtime environment. We treat all of these as fundamentally similar: they result in the execution of app-author-chosen code.
	Comments: GN : OK GA: OK GM: OK Type: INFO
	Extracted requirement:

ID	Text
46	(Note that whether an app is written in native code has no bearing on whether it is what GENIVI calls a native application, which is an app that is built into the platform, or a managed application, which is one of the user-installable apps discussed here: either may be written in either native code or an interpreted/JITted environment.)
	Comments: GA: Good paragraph but I would consider if the words "compiled code" are any help here. PW: I suspect that ‘compiled code’ was chosen against, because a large variety of programming languages are ‘compiled’, even if the end result is managed or native code. The term ‘native code’ is loosely defined on line 40 — does the definition need to be bolstered?
	Extracted requirement:

ID	Text
50	The app framework must be capable of running native-code (C or C++) executables.
	Comments: GA: Policy decision? If we're aiming for one framework to rule them all, I guess I agree. If we are aiming for a shared requirement set for many different environments, it might not be true. PW: The approach we’ve taken is that applications written in interpreted or managed languages are treated as the combination of their interpreter and the code, where the interpreter is invariably native code. So native code support is always required. However, this would be an unnecessary complication on systems which only have managed/interpreted apps. This could be changed to be a policy detail; I don’t know what the consequences would be for the rest of the document. GN : OK
	Extracted requirement:

ID	Text
51	The app framework must be capable of running programs that require an interpreter/JIT-based runtime environment such as Java or Python. It may require that the runtime environment provides suitable library functionality to work with the framework (for example, if the framework uses D-Bus for IPC, then it does not need to support runtime environments that do not have a D-Bus implementation or binding).
	Comments: GA: See previous GN : OK
	Extracted requirement:

ID	Text

56	• The app framework must be capable of running programs that run in a HTML5 runtime environment: in other words, it must be possible to package a web application into a form suitable to be an app bundle. The entry points to an app might include GUIs and/or background services (agents, daemons).
	Comments: GA: See previous GN : OK
	Extracted requirement:

ID	Text
59	• It must be possible for an app to contain zero or more GUI entry points. Each of these Application framework concept document might appear in menus (see App launching) and/or be available for launching by other means (see Document launching, URI launching, Data sharing).
	Comments: GA: See below
	Extracted requirement:

ID	Text
64	• It must be possible for an app to contain zero or more background services with no GUI, which can be launched for purposes such as Data sharing. For example, a search provider for a global search feature similar to GNOME Shell search or Unity Lenses, such as the one described in Apertis Global Search design, might be implemented in this way.
	Comments: GA: So the definition is that these are not entry-points. Entry-points require a GUI is the definitionblocked URL PW: No, the definition on line 59 is that entry points may be GUIs or background services. The definition on line 26 does not contradict this, but could be clarified as it currently sounds like an entry point must have a GUI. That's not the case. PW: Note that we could cross-reference this to ~line 28.
	Extracted requirement:

ID	Text
69	• It must be possible for the GUIs and background services to be implemented by the same executable(s) run with different options, or by separate executables.
	Comments: GA: OK GM: OK
	Extracted requirement:

ID	Text
71	Some vendors might decide to restrict the apps available in their app stores to have at most one executable, or to have at most one GUI and one non-GUI executable, but that is a policy decision by those vendors and should not be reflected in the more general app framework.
	Comments: GA: OK GN : OK GM : OK Type: INFO

	Extracted requirement:

ID	Text
75	Each bundle should have bundle metadata to represent the app in situations like an app store, a system settings GUI or a prompt requesting app permissions.
	Comments: GA: OK GN : OK GM : OK Type: Requirement
	Extracted requirement:

ID	Text
77	• As a minimum, this metadata should include a globally unique identifier, an icon, and an international (English) name and description.
	Comments: GA: OK GN : Need to define a minimum set as mandatory requirement including privileges/permissions Type: REQ
	Extracted requirement:

ID	Text
79	• Additionally, app bundles should be able to contain translations (localization) which replace the international name and description, and any other fields that are marked as translatable (internationalization), when displayed on devices configured for a specific language and/or country.
	Comments: GA: OK GN : OK GM : OK Type : REQ
	Extracted requirement:

ID	Text
83	• The metadata fields in an entry point should be in line with what is typically present in other interoperable package metadata specifications such as freedesktop.org AppStream and the parts of Android manifests that do not relate to a specific <activity>.
	Comments: GA: OK GN: OK
	Extracted requirement:

ID	Text

87	• The base set of metadata fields should be standardized, in the sense that they are described in a vendor-neutral document shared by all GENIVI vendors and potentially also by non-GENIVI projects, with meanings that do not vary between vendors. For example, AppStream XML would be a suitable implementation.
	Comments: GA: OK GN:OK Type:REQ
	Extracted requirement:

ID	Text
91	• We anticipate that vendors will wish to introduce non-standardized metadata, either as a prototype for future standardization or to support vendor-specific additional requirements. It must be possible to include new metadata fields in an entry point, without coordination with a central authority.
	Comments: GA: OK GN:OK Type:REQ
	Extracted requirement:

ID	Text
95	• For example, this could be achieved by namespacing new metadata fields using a DNS name (as is done in D-Bus), namespacing them with a URI (as is done in XML), or using the X-Vendor-NewMetadataField convention (as is done in email headers, HTTP headers and freedesktop.org .desktop files).
	Comments: GA: Discussion point. Shall we define this and if so what to choose? PW: It should be standardised if and only if the metadata format is standardised, as discussed on line 87. If so, we recommend AppStream XML, plus ancillary files for custom metadata. GA: OK. I propose we make AppStream XML a *recommended* format. Are there any other format proposals? PW: None from me. For reference, here's the Apertis specification for it: https://appdev.apertis.org/documentation/bundle-spec.html#bundle-metadata
	Extracted requirement:

ID	Text
99	Apps are expected to be numerous. • The app framework must be designed such that it does not need to place an arbitrary limit on the number of apps installed on the system, as long as their total size on storage (flash) fits within the available space.
	Comments: GA: OK GN: OK Type: REQ
	Extracted requirement:

ID	Text

103	• The app framework must be designed such that it does not need to place an arbitrary limit on the number of apps running at the same time, as long as their total size in RAM fits within the available space.
	Comments: GA: OK GN: OK GM: OK Type: REQ
	Extracted requirement:

Data management

ID	Text
107	The app framework must provide a location where app programs can write their private data.
	Comments: GA: I would rephrase "location" to mechanism. There are several ways to do this, one would be through a defined interface, like GENIVI Persistence Client Library GN : Is this expected to have a separate set of API's. PW: We used ‘location’ because Apertis does not require use of a persistence library (robustness is left to the file system and kernel to implement), but it could be rephrased to ‘mechanism’. The only API defined here is the existence and path of the shared location.
	Extracted requirement:

ID	Text
109	Open question: is this in-scope for the app framework, or is there some other platform component that does it?
	Comments: GA: The deep implementation of persistence must be a platform issue. The API for applications should be defined here. We must discuss the solution and the app-to-platform interface • if every app interaction shall be over IPC for example, then some kind of persistence daemon is required? GN : Since persistence management from platform takes care of this. In my opinion this is a requirement for platform. but if some filtering to be done about the api's exposed, then its need to be handled appropriatelyPW: If the API for this needs to be defined in the application framework design, then I suggest defining the private data location for each application, which it can then pass to the persistence library. I am not sure how apps interacting over IPC (or not) fits in with this — what is your query?
	Extracted requirement:

ID	Text
111	The framework should provide a location that is treated as private data in which to store cached data, defined as data that can be recovered in a straightforward way by downloading it from the Internet or computing it from non-cached data.
	Comments: GA: OK GN: should be ok.
	Extracted requirement:

ID	Text

114	• The framework may delete files from the cached data area at any time to free up storage space, and apps should be written to expect this.
	Comments: GA: OK. (This is a requirement on the framework but also needs to also go into app writing guidelines) GN : OK. GM : OK
	Extracted requirement:

ID	Text
116	• For app author convenience, the framework may also provide conventional locations for other sub-categories of private data such as configuration (data that has a useful default, but can be reconfigured by the user, and whose deletion would be considered to be data loss) and state (data with no useful default, whose deletion would likewise be considered to be data loss).
	Comments: GN : OK
	Extracted requirement:

ID	Text
121	The app framework must provide a mechanism by which an app program's private data can all be deleted by another system component, for example as part of removal or a factory reset.
	Comments: GN : Ok as requirement. GA: OK
	Extracted requirement:

ID	Text
124	The app framework should provide a mechanism by which all app programs' private data can be deleted in a single operation during a factory reset, so that the factory reset procedure does not need to enumerate app programs and iterate through them.
	Comments: GA: OK
	Extracted requirement:

ID	Text
127	Deleting per-user data and per-device data during a factory reset is also anticipated to be necessary, but is outside the scope of this framework.
	Comments:
	Extracted requirement:

Sandboxing and security

ID	Text
130	App processes should run in a sandbox which partially isolates them from the rest of the system.
	Comments: GA: OK GN : OK GM : OK Type : Info ?
	Extracted requirement:

ID	Text
132	We anticipate that each app bundle will act as a security domain, similar to the concept of an origin on the Web: in other words, there is a security boundary between each pair of appbundles, but for simplicity there is no privilege boundary within an app bundle (for example between two programs in the same app bundle).
	Comments: GA: OK GN: OK GM: OK
	Extracted requirement:

ID	Text
136	Each app is assumed to store private data which is specific to that app. On a multi-user system, this private data is also specific to a user: in other words, there is one private data location per (app, user) pair.
	Comments: GA: The later descriptions are clearer. I might suggest rewriting this first paragraph. GN: same as above Type : REQ
	Extracted requirement:

ID	Text
139	• Any data with this access model is considered to be private data, whether it is in files directly written by the app, files written by platform libraries used by the app, or other data stored on behalf of the app by platform services (for example accessed via interprocess communication).
	Comments: GN : Is this list final or only an example? PW: It's meant as an example. Type :
	Extracted requirement:

ID	Text
143	• Private data availability: when a specific user runs a program that is part of a specific app, that program can read and write the data owned by that (app, user) pair.
	Comments: GN :OK type : REQ

	Extracted requirement:

ID	Text
145	• Private data confidentiality and integrity: an app must not be able to read, add, change or delete data owned by a different app and the same user without the other app specifically sharing it. The program must also not be able to read, add, change or delete data owned by the same app but a different user.
	Comments: GN :OKtype : REQ
	Extracted requirement:

ID	Text
149	Note that the App confidentiality requirement below imposes a stronger requirement than this: the first app must not even be able to know that the second app's private data exists.
	Comments: GN :OKtype : REQ
	Extracted requirement:

ID	Text
152	Some categories of data might be specific to a single app but common to all users. We call these per-app data.
	Comments: GN :OKtype : Info
	Extracted requirement:

ID	Text
154	• The app framework may have support for per-app data. If it does, the availability, confidentiality and integrity requirements are analogous to those for private data. The per-app data is considered to be jointly owned by all users, therefore there is no expectation of confidentiality or integrity for the per-app data of programs from the same app bundle running as different users. Some categories of data are not necessarily specific to a single app; instead, they might be shared between all apps. We call these per-user data. For example, the user's address book (contacts) and the user's calendar (appointments) might be among these categories.
	Comments: GN :OK
	Extracted requirement:

ID	Text
162	• Any data with this access model is considered to be per-user data, whether it is in files directly written by multiple apps, files written by platform libraries used by multiple apps, or other data stored on behalf of multiple apps by platform services (for example accessed via inter-process communication).

	Comments: GN : Is this list final or only set of way of handling? PW: It's an example. Type :
	Extracted requirement:

ID	Text
166	• We anticipate that in practice, per-user data would most commonly be kept outside apps' sandboxes and accessed via inter-process communication to a shared service. For example, Android contacts provider services, GNOME evolution-data-server and KDE Akonadi all use this model for address books.
	Comments: GN : OK
	Extracted requirement:

ID	Text
170	• User data availability (read): the apps that require access to this per-user data must be able to read it. For example, a messaging application might require access to the address book so that it can read the thumbnail photos representing contacts and display them in its user interface.
	Comments: GN : OK
	Extracted requirement:

ID	Text
174	• User data availability (write): the apps that require write access to this per-user data must be able to add, change and delete it. For example, a messaging application might require write access to the address book so that it can add contacts' instant messaging addresses to it.
	Comments: GN: OK GA: "Write" is general. As requirements go, additional precision might be needed, e.g. delete != modify?
	Extracted requirement:

ID	Text
178	• User data confidentiality with least-privilege: an app must not be able to read per-user data without user consent, other than what that app needs to carry out its normal function. For example, a compromised messaging app would still be able to read the address book until the compromise was somehow detected, but would not be able to read (for example) the user's appointments calendar.
	Comments: GN : OK type : REQ
	Extracted requirement:

ID	Text

183	• User data integrity with least-privilege: an app must not be able to modify per-user data without user consent, other than what that app needs to carry out its normal function. For example, a compromised messaging app would still be able to modify the address book until the compromise was somehow detected, but would not be able to modify the user's appointments calendar.
	Comments: GN : OK
	Extracted requirement:

ID	Text
188	Some categories of data are not necessarily specific to a single app or to a single user; instead, they might be shared between all apps and between all users, like Android's /sdcard. We call these per-device data.
	Comments: GN: OK
	Extracted requirement:

ID	Text
191	• The app framework may have support for per-device data. If it does, the availability, confidentiality and integrity requirements are analogous to those for per-user data, except that there is no expectation of confidentiality or integrity for per-device data. The user might install a malicious app that has been written or modified by an attacker, or the user might install an app with a security flaw that leads to an attacker being able to gain control over that app (referred to below as a compromised app). Either way, the attacker is assumed to be able to execute arbitrary code in the context of that specific app.
	Comments: GN: OK
	Extracted requirement:

ID	Text
198	• The requirements stated above for private and user data confidentiality and integrity mitigate this attack by restricting what the malicious or compromised app can do.
	Comments:
	Extracted requirement:

ID	Text
200	• App integrity: a malicious or compromised app must not be able to modify the executables and static data of other apps.
	Comments: GN : OK type : REQ
	Extracted requirement:

ID	Text
202	• App confidentiality: in general, a malicious or compromised app must not be able to list the other apps that are running on the system or the other apps that are installed, either by their bundle names, by their entry points, or by inferring their presence from private or per-app data that they have written. Both are potentially sensitive information that could be used to "fingerprint" a particular user or class of users (for example customers or employees of a particular organization).
	Comments: GN:OK type : REQ
	Extracted requirement:

ID	Text
208	• Note that if an app has written per-user data or per-device data, then it has potentially given up its own app confidentiality, in the sense that a malicious or compromised app could potentially identify it from the per-user or per-device data that it has written out. We recommend minimizing the number of apps able to write per-user and per-device data for this reason, and preferring to use content selection, document launching and data sharing to satisfy the use-cases for which other platforms would use a per-device filesystem.
	Comments: GN: OK
	Extracted requirement:

ID	Text
215	• Similarly, in general an app must not be able to communicate with other apps without user consent. Controlled exceptions to this general rule might exist for use cases such as data sharing.
	Comments: GN: OK type: REQ GA: Needs some more details. Eg. is (direct) communication between apps even possible? User consent is a policy question? PW: We could expand on this a bit, but since this section is about file system access, it would be a bit distracting to devote a lot of discussion to inter-app communication. The idea in Apertis is that apps cannot talk directly to each other — any communication is mediated via an IPC service for ‘data sharing’ which can enforce permissions on which apps can talk to each other. These permissions might involve user consent (the precise policy is, as you say, a question of vendor policy). GA: OK, I think that makes sense.
	Extracted requirement:

ID	Text
218	• System integrity: a malicious or compromised app app must not be able to violate the integrity of the system as a whole (for example by modifying the executables or static data of the system, or by altering the system's idea of what is a trusted app source).
	Comments: GN : OK type : REQ
	Extracted requirement:

ID	Text
221	Resource limits: A malicious, compromised or buggy app might use more than its fair share of system resources, including CPU cycles, RAM, storage (flash) or network bandwidth.
	Comments: GN : OKtype : REQ
	Extracted requirement:

ID	Text
223	• Each app must have its own limit for these various metrics, for example by using cgroup resource controllers.
	Comments:
	Extracted requirement:

ID	Text
225	• If this limit is exceeded, the vendor may choose how to respond to this. Options include killing or freezing the app, rate-limiting requests, denying requests, and /or reporting the app to the app-store as potentially malicious.
	Comments:
	Extracted requirement:

App permissions

ID	Text
229	A very simple app, for example a calculator or a simple to-do list, might not need to do anything other than the operations allowed to all apps: display a GUI when launched, run code in a sandbox, store its own private data up to some reasonable limit, and so on.
	Comments:
	Extracted requirement:

ID	Text
232	To carry out its designed purpose, a more complex app might need permission to carry out actions that can compromise confidentiality (user privacy), integrity, or availability (the absence of denial-of-service). For example, a more elaborate to-do list app might be able to synchronize the to-do list to a cloud service, requiring it to have Internet access which would make it technically able to copy whatever data it can read to a location not under the user's control; it might ask to read the user's geographical location, to provide locationbased reminders; and it might support attaching photos to its to-do items, requiring it to read files that are not its private data.
	Comments:
	Extracted requirement:

ID	Text
240	Some permissions have technical constraints that makes it impractical to request user permission before they are used. For example, one possible permission flag is "has unrestricted Internet access", which might be used for a voice-over-IP client app. To support this control, the life-cycle manager would need to launch the app program with unrestricted Internet access either allowed or forbidden: it cannot be adjusted later.
	Comments: GA: "before they are used" really means "on-demand" or "interactively"? Internet access, system might also ask consent on first use. PW: Yes. In this case, we're talking about situations where an entire session in an app would need to be torn down and recreated in order to change its behaviour after being granted/denied permission. [App FW telco - 15-11-2016] Its agreed to change to "on demand". Provide possibility for having permissions before launch and after launch.
	Extracted requirement:

ID	Text
245	• App bundles must be able to specify permissions without which they will not work, given in bundle metadata.
	Comments:
	Extracted requirement:

ID	Text
251	• The user might be asked whether to grant those permissions on installing that app bundle or on launching any entry point from that bundle, or the framework might automatically grant certain permissions based on approval from an app-store curator without user interaction. Some permissions can usefully be granted or denied at runtime. For example, address book access on Android works like this: the permissions framework can be configured to prompt the user on each attempted access.
	Comments:
	Extracted requirement:

ID	Text
254	• Operations that cross a privilege boundary between processes should include a step where a platform security framework is queried, to check whether the user's permission for the privileged action has been given. This should have at least three possible policy outcomes: allow, deny, or ask the user.
	Comments:
	Extracted requirement:

ID	Text
258	Some operations that cross privilege boundaries naturally include an opportunity for the user to reject the operation. To minimize driver distraction, the system should provide that opportunity instead of having a separate permission prompt.
	Comments:
	Extracted requirement:

ID	Text
261	• If an operation will naturally result in the user being prompted for a decision of some sort, there should not be an additional prompt for whether to allow the action. Instead, the user can indicate lack of consent by declining to make the requested decision. For example, content selection could use this approach: the user implicitly indicates consent to open or attach a file by selecting it, or indicates lack of consent by cancelling the file-selection dialog.
	Comments:
	Extracted requirement:

ID	Text
267	• The framework might require that particular privilege-boundary-crossing operations are declared in advance even though they imply an opportunity for the user to reject the operation, for example if those operations are considered to be particularly sensitive or vulnerable to social engineering attacks. If it does, then it may make attempts to invoke those operations fail unconditionally, as if the user had canceled them but without prompting the user at all.
	Comments: GA: I would say framework **shall** require all privilege-boundary-crossing operations to be declared... (why not?). I mean they need to be declared - not every operation might be in the category that requires user acceptance. User acceptance and declaration are orthogonal concepts in my view of this. PW: I see no problem with declaring them all in advance. As you say, declaring which permissions an application *might* request is orthogonal to whether those permissions are actually granted at request time. [App FW telco - 15-11-2016] : agreed
	Extracted requirement:

ID	Text
273	• Operations that cost money might be considered to be particularly sensitive — for example, a parent installing apps on behalf of a child is likely to want to prevent them — so the framework implementor might wish to ensure that operations like "send SMS" and "make in-app purchases" must be declared in advance.
	Comments: Type: Information
	Extracted requirement:

ID	Text
277	• Access to online accounts (such as social media) might be considered particularly susceptible to social engineering (since a user might not recognize when a request to fill in their social media account/password is or isn't legitimate), so the framework implementor might wish to ensure that operations involving these accounts must be declared in advance.
	Comments: GA: As indicated above, I would change "declared in advance" to "require user acceptance" or similar. Discuss and improve. PW: I think we can treat declaring permissions as orthogonal to user authorisation of an app requesting those permissions in this case too. In this case, an example permission would be “let this app use my Facebook account to upload a photo”. If the permission is declared for the app, there are several policies which could be used at the time the app actually tries to start the upload: unconditionally allow the request, unconditionally deny the request, always ask the user, or ask the user if this is the first time the permission has been requested and use the answer from last time otherwise. I would agree with you, but I suspect that we should leave the exact policy to the vendors. [App FW telco - 15-11-2016] similar to ID-267

	Extracted requirement:

App launching

ID	Text
283	A bundle may contain zero or more entry points. These are typically started from a launcher, which might take the form of a home screen, main menu or application list.|
	Comments: GA: OK, although I think it's a little unclear with reference to the defintion. OpenDocument(mimetype) is also an entry point right? At least if it brings up a GUI, but it won't be on the home screen. Only some entrypoints will? PW: Correct, that's why it says 'typically' rather than 'always'. We could rephrase this to 'for example, these might be started from…'? [App FW telco - 15-11-2016] : Have the entry point topic collated together and make it explicit.
	Extracted requirement:

ID	Text
285	• A launcher must be able to list all of the visible, available entry points in any installed bundle, together with enough metadata to display them in its menus. As a minimum, this would typically include a multilingual/localized name and an icon. Other metadata fields, such as categories, could be useful or unnecessary depending on the launcher's UX.
	Comments: GA: Only those entrypoints that are not OpenDocument() type... PW: There seems to be some confusion here over entry points which handle opening content by its content type (equivalently, its MIME type). Any entry point, including those which are listed in the launcher, can declare that it handles a list of content types. An entry point does not have to be listed in the launcher to declare that it handles a list of content types, either. For example, the Apertis music application has four entry points: three (artists, songs, albums) are listed in the launcher and don't handle opening content; and a fourth (the 'now playing' view) is not listed in the launcher but *does* handle opening content (audio/mpeg, etc.). Different entry points within the same applications bundle can advertise different lists of content types they handle. Phrased differently, the set of entry points which handle opening content and the set of entry points listed in the launcher do not have to be disjoint or equal. [App FW telco - 15-11-2016] : As agreed above wrt to entry points.
	Extracted requirement:

ID	Text
290	• The metadata fields in an entry point should be in line with what is typically present in other interoperable menu-entry specifications, such as freedesktop.org . desktop files or the <activity> element in Android manifests.
	Comments: GA: OK
	Extracted requirement:

ID	Text
293	• The base set of metadata fields should be standardized, in the sense that they are described in a vendor-neutral document shared by all GENIVI vendors and potentially also by non-GENIVI projects, with meanings that do not vary between vendors. For example, .desktop files would be a suitable implementation.

	Comments: GA: First part is OK but decision needs discussion. Does .desktop imply also the syntax? E.g. ini-file style...? PW: .desktop implies the syntax, standard field names and semantics, and rules for defining custom fields. http://standards.freedesktop.org/desktop-entry-spec/latest/ GA: OK. Let's make .desktop format a recommended format. Are there any other proposals? PW: I have no other proposals. For reference, here's the Apertis specification for it: https://appdev.apertis.org/documentation/bundle-spec.html#entry-points [App FW telco - 15-11-2016] : agreed.
	Extracted requirement:

ID	Text
297	• We anticipate that vendors will wish to introduce non-standardized metadata, either as a prototype for future standardization or to support vendor-specific additional requirements. It must be possible to include new metadata fields in an entry point, without coordination with a central authority.
	Comments:
	Extracted requirement:

ID	Text
301	For example, this could be achieved by namespacing new metadata fields using a DNS name (as is done in D-Bus), namespacing them with a URI (as is done in XML), or using the X-Vendor-NewMetadataField convention (as is done in email headers, HTTP headers and freedesktop.org .desktop files).
	Comments: GA: Yes. Needs discussion, to **decide** something. GA: Did anyone make notes from previous call discussion? I don't understand what people prefer here - make proposals, vote, and reach a conclusion. PW: Guru said he was taking notes, but they haven't materialised here yet. From what I remember, this question is determined by the choice of metadata format for entry points. So since we've decided to go with .desktop files, the namespacing must use the X-Vendor-NewMetadataField scheme. [App FW telco - 15-11-2016] : Agreed.
	Extracted requirement:

ID	Text
305	• Because of the requirement that ordinary app bundles are not allowed to enumerate other app bundles or entry points, if a launcher is implemented as a user-installable app bundle (as is sometimes done on Android), it must have a special permissions flag allowing it to carry out that restricted action.
	Comments: GA: OK. Not sure if this is a requirement from any OEM (to have a use-installable launcher). It could be implemented using an app permission, or by having "native" privileged APIs not available to any app. [App FW telco - 15-11-2016] PW : e.g. third party call on system bus may not be allowed.As this needs fine grain access control. This can complicate or make things complex. GA : Agreed to go ahead with the existing requirement
	Extracted requirement:

ID	Text
309	Some entry points might be flagged to not be visible in menus. For example, an app that is a viewer for some file type such as PDF might register itself as a handler for files of that type, but might not have anything useful to do if it appears in menus otherwise.

	Comments: GA: This seems to be a policy question. Requirement is only to make this possible? GA: (I need another discussion on definition of entry point, visible vs background apps etc...) PW: It's not really a policy question, it's more of an implementation question for the app developer. In the case of an entry point for handling PDF files, if the app doesn't do anything unless given the path to a PDF file, it's not going to make sense to display that entry point in menus on any vendor's system. JK: OK for another use case (e.g. hide an app icon from the launcher w/o removing) but agreed w/ Gunnar. The use case above is based on the assumption that an entry point to handling PDF shall be defined as other entry points having an icon, which needs discussion. [App FW telco - 15-11-2016] : This should be left to the policy handling
	Extracted requirement:

ID	Text
312	• Entry point metadata must indicate whether the entry point is to be visible in menus.
	Comments:
	Extracted requirement:

ID	Text
313	• The mechanism used by the launcher to list entry points may either include or exclude invisible entry points. If it does include those entry points, it must also provide the launcher with an indication that they are to be made invisible.
	Comments:
	Extracted requirement:

ID	Text
316	When the user selects an entry point, the expectation is that the program that implements that entry point should be launched.
	Comments:
	Extracted requirement:

ID	Text
318	• If the program that implements the entry point is not already running, the system must run it. (See also life-cycle management.)
	Comments:
	Extracted requirement:

ID	Text
320	• The program might implement more than one entry point. It must be told which entry point was launched, for example via command-line arguments or an inter-process communication call.
	Comments:
	Extracted requirement:

ID	Text

323	We do not anticipate that ordinary (non-launcher) app bundles would have a reason to launch specific entry points in this way: we expect that if app bundles need to communicate, they will do so via document launching, URI launching or data sharing. This does not preclude one executable in a bundle from running another executable in the same bundle directly.
	Comments:
	Extracted requirement:

ID	Text
328	• Open question: Do ordinary app bundles need to be allowed to launch other bundles' entry points by name? If so, why?
	Comments:
	Extracted requirement:

ID	Text
330	• Android does allow this, but Android does not appear to provide app confidentiality.
	Comments:
	Extracted requirement:

ID	Text
331	• One possible use-case for a program launching a program outside its bundle would be to bring up the system settings. For example, Android apps that make use of location services often have a shortcut button to bring up the Location panel in the built-in Settings app, because the user-installable app would not be able to enable location itself, but its author wishes to make it easy for the user to do so.
	Comments:
	Extracted requirement:

ID	Text
336	However, a vendor-specific Settings app is part of the platform rather than being a user-installable app bundle, so the constraints applying to it and the APIs that can be used with it do not have to be the same as for app bundles.
	Comments:
	Extracted requirement:

ID	Text
339	This would also be easy to implement without launching the Settings app by name: the built-in Settings app could register for URI launching as the launcher of a URI scheme, similar to the way the iOS Settings app used to register the prefs URI scheme, and the user-installable app could launch a URI of that scheme.
	Comments:
	Extracted requirement:

Document launching

ID	Text
344	Some app entry points will provide handlers for particular file types.

	Comments: GN :
	Extracted requirement:

ID	Text
345	• An entry point must be able to identify the file types that it can receive. For example, a document viewer might register itself to receive Microsoft Word documents, Open Document Text files, and PDFs.|
	Comments:
	Extracted requirement:

ID	Text
348	• We recommend that these are identified via IETF media types (also known as content types or MIME types), because the IETF media types are an extensible standard, are ubiquitous in existing operating system environments such as Windows, OS X, Android and freedesktop-based environments such as GNOME, and are part of key Internet technologies such as HTTP and email.
	Comments: GN : This needs to be translated into requirement. GA: IETF, but a human readable name also needed right? PW: We assume that there is a separate mapping from IETF media types to human readable names for the file types. This exists as the Shared MIME-Info database on Linux systems. [App FW telco - 15-11-2016] : Agreed
	Extracted requirement:

ID	Text
353	• The app framework must be able to identify the format of a file on secondary storage (flash), for example via its extension or "magic number". Unidentified files must be considered to have a documented generic format, for example application /octetstream in the IETF media type system.
	Comments:
	Extracted requirement:

ID	Text
357	• Open question: it has been suggested that app-bundles should be able to define their own new file types. Is this a requirement?
	Comments: GN :Could you provide a use case why such a requirement is needed? PW: This would be needed if an app needed to save files in a format which does not currently have a standard content-type, and then have those files launch it when they are selected in the file manager. i.e. The cases where an app needs to register a new entry in the MIME magic database. \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ GA: IMHO the possibility to do it is a requirement. To allow it or not is OEM policy. PW: Agreed, if we want to allow it at all (see the discussion below). JK: Agreed [App FW telco - 15-11-2016] : Agreed.
	Extracted requirement:

ID	Text
359	• This requirement seems unwise from the point of view of system integrity: if an appbundle can define its own file types with their own extensions and/or "magic numbers", then it can introduce a conflict with other app-bundles or even alter the interpretation of existing files.

	Comments: GN :Could you provide a use case why such a requirement is needed? PW: The use case for preventing this is to prevent applications from assigning conflicting content types to existing files. A file can only have one content type, and if the MIME magic database is coerced into assigning the wrong type for it, the file will end up being opened with the wrong application. \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ GA: Let's dig deeper into the risks/opportunities of this. Mitigation for conflicts? Do you mean can be defined at run-time or can be defined in manifest? PW: MIME type associations would have to be defined in manifests. They cannot be defined at application run-time because the system needs to know the magic bytes for a file type before it can work out which application to launch it with. However, manifests are trusted to have been audited when the application was submitted to the vendor, so we should be able to trust that each manifest only defines MIME types for file formats under control of that application's author. And hence we should be able to guarantee there are no conflicts. So overall, I think I disagree with this part of the document (that allowing new MIME types to be defined is a risk), and think we should be able to allow it safely. [App FW telco - 15-11-2016] : GA : Rephrase as a recommendation to the policy maker. PW - Agreed.
	Extracted requirement:

ID	Text
363	If this is implemented at all, we recommend that it should be tightly controlled by app-store curators.
	Comments: GA: Supplementary text / guideline. OK.
	Extracted requirement:

ID	Text
365	• Choice of document handler: When a file is activated (for example by tapping its icon) from a non-app context such as the home screen, the app framework must locate the entry points that are able to handle that file. It must either choose one of those entry points for use, or prompt the user to choose one.
	Comments: GN : Ok
	Extracted requirement:

ID	Text
369	• When a file is activated from the context of an app (the initiating app), for example if the user activates an attachment in an email app, the app framework must behave similarly. It may opt to follow a different policy for choosing the correct entry point in this case; for example, it might prompt the user for confirmation even if there is only one possible handler.
	Comments: GN : OK as Requirement
	Extracted requirement:

ID	Text
374	• System vendors must be able to force a particular app to handle particular file types. For example, a vendor might wish to make their video player handle all videos.
	Comments: GN : OK as requirement.
	Extracted requirement:

ID	Text

376	• If no handler is available for the selected file type, the app framework should arrange for a suitable fallback to be displayed. For example, it might show an error message, or it might launch its app store user interface with a search query for the handlers for that file type.
	Comments: GN : Error handling requirement and choice should be left to vendor/OE specific handling PW: I think the intention here is that the error handling *is* left as a policy decision for the vendor. [App FW telco - 15-11-2016] : Vendor specific. OK
	Extracted requirement:

ID	Text
380	• No feedback to initiator: It should do this itself or by interacting with other system components instead of feeding back an error code to the initiating app (if any), because otherwise the initiating app would be able to use this as an "oracle" to gather information about the set of installed app bundles.
	Comments: GN : Ok as requirement.
	Extracted requirement:

ID	Text
384	• User confirmation: If exactly one handler is available for the selected file type, the app framework may launch it directly, or ask the user for confirmation. If the user cancels a request for confirmation, the app framework should neither launch the handler nor feed back an error code to the initiating app.
	Comments: GN : Error handling requirement and choice should be left to vendor/OE specific handling PW: All behaviour apart from what is required here is left as a policy decision for the vendor. The features mandated in this line are needed for security — not feeding back an error code is required to prevent acting as an oracle for the number of apps installed which can handle the given content type, which is an indicator of which apps the user has installed (especially for uncommon content types). GA: The possibility of user confirmation seems to be repeated from previous text. Simplify? PW: Are you referring to line 369? I don't think there's any harm in explicitly repeating the possibility here. [App FW telco - 15-11-2016] OK
	Extracted requirement:

ID	Text
388	• If more than one handler is available for the selected file type, the app framework may launch a preferred handler directly, or ask the user to make the choice. If the user cancels a request for app choice, the app framework should neither launch a handler nor feed back an error code to the initiating app.
	Comments: GN : Error handling requirement and choice should be left to vendor/OE specific handling PW: See my comment for ID 384. [App FW telco - 15-11-2016] OK
	Extracted requirement:

ID	Text
392	• The app framework must arrange for the file's content to be made available in a location where the chosen app can read it (see sandboxing and security).
	Comments:
	Extracted requirement:

ID	Text
394	• If the program that implements the entry point is not already running, the system must run it. (See also life-cycle management.)
	Comments: GN : Could this be a vendor specific. It could be that system has not reached a state / condition which is a prerequisite for the launch an any app. PW: We were assuming that the system would either pause the app launch until it is in a state to launch apps, or not allow entry points to be spawned until that time. i.e. That it can synchronise on its own state.
	Extracted requirement:

ID	Text
396	• The program must be told that it was launched to open a file, and given the filename of the file to open, for example via command-line arguments or an inter-process communication call. The filename that it is given might differ from the original file that was activated, for example if the file had to be copied or linked across a privilege boundary to be made available in the program's sandbox. The program must be able to distinguish between this action and ordinary app launching.
	Comments: GN : What could be a requirement for App FW? PW: I suggest the final sentence becomes the requirement. [App FW telco - 15-11-2016] : OK
	Extracted requirement:

ID	Text
402	• Programs should be careful not to treat documents received in this way as executable code, or assume that the source of the document is trustworthy. For example, macro languages in "office" document formats should be disabled, and if arbitrary code execution in a program can be triggered by a malformed document, this should be considered to be a security vulnerability.
	Comments: GN : OK.
	Extracted requirement:

ID	Text
407	• We do not anticipate a need for the initiating app to be able to influence the choice of launched app.
	Comments:GN :OK, what requirement can be framed from this? PW: I don't think there is a concrete requirement in either direction here. [App FW telco - 15-11-2016] : Info text
	Extracted requirement:

ID	Text
409	• If the initiating app could influence the choice of launched app, a malicious app could potentially use this to break or undermine app confidentiality. For example, suppose org.example.Secret opens .secret files. If the app com.example.Spy wanted to determine whether org.example.Secret was installed, it could register an entry point com.example.Spy.SecretHandler which also opens .secret files, create a .secret document, and launch that document specifying org.example.Secret and com.example.Spy.SecretHandler (in that order) as the preferred handlers. If com.example.Spy.SecretHandler was launched, then com.example.Spy could be sure that org.example.Secret was not installed. Conversely, if com.example.Spy.SecretHandler was not launched, then com.example.Spy could infer that org.example.Secret was likely to be installed.

	Comments: GN : OK
	Extracted requirement:

ID	Text
420	Apertis Content Hand-over Use Cases contains some similar requirements-capture that was carried out for the Apertis platform.
	Comments: GN:OK
	Extracted requirement:

URI launching

ID	Text
423	Some app entry points will provide handlers for particular URI schemes such as https, mailto or skype.|
	Comments: GN:OK
	Extracted requirement:

ID	Text
425	• file URIs must not be included in this mechanism. Instead, they should be decoded into filenames and processed via document launching.
	Comments: GN:OK
	Extracted requirement:

ID	Text
427	• An entry point must be able to identify the URI schemes that it can receive. For example, a multi-protocol voice-over-IP client might support receiving sip and xmpp URIs.
	Comments: GN : Can this be a requirement? PW: Yes. In general I think that anything phrased as ‘must’ in the document should become a requirement. [App FW telco - 15-11-2016] : GN : Its a part of manifest field, PW - Yes its a std key
	Extracted requirement:

ID	Text
430	• When a URI is activated, the app framework must locate the entry points that are able to handle that URI and choose one for launching, much like file type handling. The same points about choice of handler, user confirmation, and lack of feedback to the initiating app apply equally here.
	Comments:
	Extracted requirement:

ID	Text
434	• As with URI schemes, system vendors must be able to force a particular app to handle particular URIs. For example, a vendor might wish to make their built-in web browser handle all http and https URIs.

	Comments:
	Extracted requirement:

ID	Text
437	• If the program that implements the entry point is not already running, the system must run it. (See also life-cycle management.)
	Comments: GN : Could this be a vendor specific. It could be that system has not reached a state / condition which is a prerequisite for the launch an any app. PW: See my comments on ID 394. [App FW telco - 15-11-2016] : OK
	Extracted requirement:

ID	Text
439	• The program must be told that it was launched to open a URI, and given the URI to open, for example via command-line arguments or an inter-process communication call. The program must be able to distinguish between this action, document launching and ordinary app launching.
	Comments:
	Extracted requirement:

ID	Text
443	• As with document launching, we do not anticipate a need for the initiating app to be able to influence the choice of launched app, but system components might need to do so.
	Comments:
	Extracted requirement:

ID	Text
446	• Programs should be careful not to interpret URIs in a way that a malicious or compromised initiating app could use to violate integrity, confidentiality or availability. For example, telephone calls and text messages (SMS) could cost money, distract the driver, or divulge sensitive information to a third party. As a result, an app that acts as a tel: URI handler may respond to URI launching by offering the user a choice of actions to carry out (for example "call" and "send SMS" buttons, perhaps with a text input widget pre-filled with SMS text taken from the URI), but must not actually initiate the call or send the SMS until the user requests it.
	Comments: GN : How this can be achieved? or is this always routed through system UI. Can this be a vendor specific for how to handle? PW: I think this needs to be URI-scheme-specific for how to handle it. It could be vendor specific, but vendors will likely come up with the same approaches since the problems are fairly constrained. I don't have any more suggestions for how this could be achieved beyond what's in the examples. [App FW telco - 15-11-2016] : OK. PW : Need to extract req from this.
	Extracted requirement:

ID	Text
454	Similarly, if a URI scheme is designed in such a way that dereferencing a URI can cause content to be modified or deleted (an unsafe request in HTTP terminology), then the program interpreting the URI should ask the user before proceeding.
	Comments:
	Extracted requirement:

ID	Text
457	Apertis Content Hand-over Use Cases contains some related requirements-capture that was carried out for the Apertis platform.
	Comments:
	Extracted requirement:

Content selection

ID	Text
460	App programs might wish to interact with data stored in locations that are not naturally accessible to the app. For example, an attachment to an email would be private data for the email app as run by the user whose email account is accessing it. However, we would like to avoid such data passing through a per-device data storage area that is shared between all apps (similar to Android's /sdcard), because in practice data passed between programs will typically include sensitive data such as photos and documents.
	Comments: GN : OK type : REQ
	Extracted requirement:

ID	Text
467	The solution that is used in Apple's iOS and planned for the Flatpak system is to have an API call that creates a file-opening or file-saving dialog. While visually presented as if it was part of the requesting app, this dialog actually exists outside the app's security context (it is privileged), and it is able to browse all of the user's files. iOS calls this the Document Picker, while Flatpak calls it the Document Portal.
	Comments: GN : OKtype : INFO
	Extracted requirement:

ID	Text
472	• The app framework should provide a way to ask the user to browse for a file to open for reading, similar in principle to the conventional "Open" dialog on desktop operating systems.
	Comments: GN : OKtype : Info
	Extracted requirement:

ID	Text
475	• If the user does so, the framework must make this file available to the app program for reading.
	Comments: GN:OK type : REQ
	Extracted requirement:

ID	Text
477	• If the user cancels this prompt, the framework must indicate this to the requesting app, and must not grant it any additional access to any files.

	Comments: GN : OKtype : REQ
	Extracted requirement:

ID	Text
479	• The app framework should provide a way to ask the user to browse to a location in which to write a file, and simultaneously choose a name for that file.
	Comments: GN : OKtype : REQ
	Extracted requirement:

ID	Text
481	• As above, depending on the user's choice, the framework must either provide a way for the app to write to that location and name, or indicate cancellation and not provide any additional access.
	Comments: GN : OKtype : REQ
	Extracted requirement:

ID	Text
484	• If the user selects an existing file outside the app's sandbox, it must be overwritten atomically if the underlying filesystem supports that.
	Comments: GN : OK
	Extracted requirement:

ID	Text
486	• The app framework may provide specialized versions of this functionality for specific file types, in particular images/photos.
	Comments: GN : What do yo mean by specialized versions? what probably can be covered? PW: For example, by providing an image preview in the file selection dialogue. This can be clarified in the text. [App FW telco - 15-11-2016] : GA : Replace functionality with more explicit info. PW : OK
	Extracted requirement:

Data sharing

ID	Text
489	The system might require the ability to enumerate the implementations of a particular service or set of functionality. In this document we will refer to that set of functionality as an interface. One use-case for this is that a global search facility within the platform needs to discover a list of background services (entry points) within app bundles that can provide search results in response to user queries entered into some global search UI; for example, a Spotify client could use the search term to match artists or songs.
	Comments: GN:OK type : UseCase
	Extracted requirement:

ID	Text

495	• Suitably privileged components of the system must be able to enumerate the implementations of an interface.
	Comments: GN : OK
	Extracted requirement:

ID	Text
497	• Suitably privileged components of the system must be able to communicate with the implementations of an interface.
	Comments: GN : OK type : REQ
	Extracted requirement:

ID	Text
499	• If the system initiates communication with an implementation of an interface that is not already running, the app framework must arrange for the implementation (an entry point) to be started.
	Comments: GN : OK type : REQ
	Extracted requirement:

ID	Text
502	An app might also require the ability to enumerate the implementations of a particular interface. One example use-case here is that if an app will display a Sharing menu similar to the UX seen in Android, it needs to be able to list the apps with which files or data can be shared, in order to populate that menu. Due to the app confidentiality requirement, this should only be allowed if the interface in question is one whose implementors are aware that it will result in other apps being able to enumerate their apps. In this document we will refer to this as a public interface.
	Comments: GN : OK type : UseCase and info
	Extracted requirement:

ID	Text
509	• An app with appropriate app permissions must be able to enumerate the implementors of a public interface.
	Comments: GN : OK type : REQ
	Extracted requirement:

ID	Text
511	• Depending on the system and the interface in question, a special permission flag per public interface might be required to list the implementors, or that information might be available to every application.
	Comments: GN : OK type : REQ

	Extracted requirement:

ID	Text
514	• An app with appropriate app permissions must be able to communicate with all of the implementors of a public interface, for example via an inter-process communication channel such as D-Bus.
	Comments: GN : OK
	Extracted requirement:

ID	Text
517	• If an app initiates communication with an implementation of an interface that is not already running, the app framework must arrange for the implementation (an entry point) to be started.
	Comments: GN : similar to ID 499
	Extracted requirement:

ID	Text
520	The Apertis Interface Discovery design and Apertis Data Sharing design describe use-cases, requirements and proposed implementations for this topic in the Apertis system.
	Comments: GN : OK type : info
	Extracted requirement:

Sharing menu

ID	Text
523	One specific use-case for data sharing is a menu for sharing content with other users, for example via social media, email or real-time communications, similar to the Android Sharing menu.
	Comments: GN : OK type : UseCase
	Extracted requirement:

ID	Text
526	Two possible UXs for this facility are presented in the Apertis Sharing design. Each UX motivates rather different requirements for how this facility interacts with apps, and in particular its impact on app confidentiality.
	Comments: GN : OK type : info
	Extracted requirement:

ID	Text
529	Open question: Is this in the scope of the application framework? If it is, which UX do we intend to support?

	Comments: GN : Ux is not in the scope of App FW. However any interface that need to be exposed to UX should be taken care of. PW: UX is not in scope, indeed; but we need to know which of the UXs needs to be supported (both?), as it affects the underlying API. [App FW telco - 15-11-2016] : OK, agreed.
	Extracted requirement:

Life-cycle management

ID	Text
532	Under various circumstances (including those described in app launching, document launching, URI launching and data sharing), the system must be able to start a program provided by an app bundle.
	Comments: GN : OK GM : OK type : REQ
	Extracted requirement:

ID	Text
535	This topic overlaps with the functionality of the GENIVI Node Startup Controller, and more generally the GENIVI Lifecycle cluster. It should potentially be considered to be an orthogonal topic outside the scope of the App Framework design. Some requirements in this area are outlined here in the hope that they can be used to clarify the division of responsibilities.
	Comments: GN : OK GM : OK type : info
	Extracted requirement:

ID	Text
540	The possible states of a program in an app are as follows:
	• Not installed
	• Inactive (installed but not running)
	• Running
	• Paused The valid state transitions move linearly through that list in single steps, as follows:
	• Not installed inactive: install app bundle
	• Inactive running: start (launch), see this section
	• Running paused: pause, see this section
	• Paused running: unpause, see this section
	• Running inactive: stop (kill, terminate), see this section
	• Inactive not installed: remove app bundle

	Comments: GN : When in the Running state how the background or foreground of an app is handled? If this is left to the App how App FW will not about which apps been shown in foreground and which are in background? PW: It's assumed that the foreground/background state of an application is basically orthogonal to its inactive/running/paused state, to the extent that inactive apps have no processes running; running apps can be in the background or foreground; paused apps can be in the background or foreground (but the (paused, foreground) state probably only makes sense transitionally). Multiple apps could be (running, foreground) if the vendor supported presenting multiple apps on the screen at once (i.e. a tiling window manager). type : REQ [App FW telco - 15-11-2016] : PW : this is covered in LCM f/w, GA, GN : Its a different philosophy. Here we need it at the App Level. agreed.
	Extracted requirement:

ID	Text
552	Transitions do not skip a step: for example, a paused app process cannot be stopped without first unpausing it, and an app bundle cannot be removed until all of its processes have been stopped.
	Comments: GN : OK GM : OK type : REQ
	Extracted requirement:

ID	Text
555	Open question: some GENIVI documents have the concept of "activating" a program, which appears to be distinct from launching it. Does this correspond to selection, similar to single-clicking an icon in a desktop environment where double-clicking would cause launching; or does it represent a transition away from an intermediate state where a newly installed app is unavailable until an activation, enabling or licensing step has been performed, similar to the concept of activating a Windows installation; or is it something else?
	Comments: GA: Please reference the document in question. PW: I'm not sure which document Simon was referring to. I'll try to find out.
	Extracted requirement:

ID	Text
562	As a prerequisite for sandboxing and security, app processes must be identifiable.
	Comments: GN : OK GM : OK type : REQ
	Extracted requirement:

ID	Text
563	• The app framework must be able to start processes, either directly or by asking a separate service manager such as the Node Startup Controller to start them.
	Comments: GN : Can this be a vendor specific? PW: I suspect the mechanism for starting processes will be vendor-specific. The requirement that the app framework must be able to start processes is a requirement for all vendors. type : REQ
	Extracted requirement:

ID	Text

565	• Process tagging: each process executing code from an app bundle must be marked with the unique identifier of that bundle (for example by placing it in a suitably named cgroup or by running it under a suitable LSM context). Those processes and their child processes, whether running the same or a different executable from the app bundle or running an executable provided by the system, must not be able to enter a state where they are no longer identifiable as belonging to their bundle.
	Comments:
	Extracted requirement:

ID	Text
572	• Depending on the vendor's UX design and the app author's UX design, the entry point might start in a default state, or it might start by restoring the last-used context. The app framework should be able to send a hint that indicates which of these modes is preferred (see the section on Last-used context).
	Comments: GN: OK
	Extracted requirement:

ID	Text
576	The application launch has various interactions with the graphical user interface. See Apertis Compositor Security design for more detailed requirements-capture for the interaction between the GUI shell and apps. The Apertis design assumes that the compositor and the GUI shell are combined, as was done in Apertis' Mildenhall reference UI and in GNOME's GNOME Shell. In a system where the GUI shell and compositor are separate, those requirements should be read as being requirements for the combined system consisting of the GUI shell and the compositor.
	Comments:
	Extracted requirement:

ID	Text
583	• Processes may request that windows (surfaces, layers) are displayed. The GUI shell must be able to identify the app bundle to which a window belongs, so that it can instruct the compositor (layer manager) to display it (or not display it) according to its UX policy.
	Comments:
	Extracted requirement:

ID	Text
587	• The GUI shell must be able to identify which windows belong to the same user-facing app, so that they can be associated visually, and so that it can prevent apps from setting up misleading situations like a dialog from one app drawn over another app's window.
	Comments:
	Extracted requirement:

ID	Text

591	• The GUI shell might have an application-switcher similar to the one in Android. It must be possible to mark each app's collection of windows with a name and icon as is done in Android. This is important for the integrity of the UX — otherwise, it would be impossible for the user to tell which app is producing a given window, for example to see which app is responsible for an advertising popup (output integrity), or which app is requesting entry of a password (input integrity).
	Comments:
	Extracted requirement:

ID	Text
597	• If application launching is in progress but no window has been displayed yet, the framework must avoid focus stealing: in other words, it must ensure that input intended to go to the previous foreground window in a particular screen area is not inadvertently directed to a window presented by the newly launched application.
	Comments:
	Extracted requirement:

ID	Text
601	• One possible implementation is to disable input, send the previous app to the background, or display a placeholder while waiting for a launched app to become available, so that the app cannot appear while the user is halfway through another interaction with the previous app.
	Comments:
	Extracted requirement:

ID	Text
605	• Another possible implementation is to track whether user continues to interact with the previous app, and if they do, keep the previous app in the foreground and place the newly launched app's window in the background when it appears.
	Comments:
	Extracted requirement:

ID	Text
608	To improve perceived responsiveness, the GUI shell might display an indication that a particular entry point or app is starting.
	Comments:
	Extracted requirement:

ID	Text
610	• Startup notification (successful case): the GUI shell must be notified by the life-cycle manager when a particular entry point is starting. It must also be notified when the entry point becomes available, either explicitly (another notification from the lifecycle manager) or implicitly (a window is displayed by the appropriate app-bundle with the entry point's identifier as metadata) so that it can withdraw the indication.
	Comments:
	Extracted requirement:

ID	Text
615	• To meet the app confidentiality requirement, these notifications must not be visible to other apps.
	Comments:
	Extracted requirement:

ID	Text
617	• Startup notification (unsuccessful case): the GUI shell should be notified by the lifecycle manager when an attempt to start a particular entry point fails, so that it can withdraw the indication and display a warning instead.
	Comments:
	Extracted requirement:

ID	Text
620	• To meet the app confidentiality requirement, these notifications must not be visible to other apps. If an app program crashes or otherwise exits unexpectedly, the system might restart it.
	Comments:
	Extracted requirement:

ID	Text
623	• This must be rate-limited, to avoid infinite restart loops that could consume disproportionately many CPU cycles. For example, apps might be configured such that more than n restarts within t seconds will cause further attempts to restart the app to be abandoned. For responsiveness, we recommend that the restart counter and time are reset when the user specifically launches an entry point.
	Comments:
	Extracted requirement:

ID	Text
628	An app program might have costly graphical processing which its author wants it to stop doing while not visible.
	Comments:
	Extracted requirement:

ID	Text
630	Open question: Are these requirements regarding visibility applicable to the application framework, or to life-cycle management, or are they in the scope of the compositor or the combined system consisting of the compositor and GUI shell?
	Comments:
	Extracted requirement:

ID	Text

633	• The app framework should send a notification to the app program at each transition from one or more windows visible to no windows visible, telling it that it has been moved to the background (become invisible).
	Comments:
	Extracted requirement:

ID	Text
636	• The app may still paint its window(s) while in the background. Their new contents must be used in any context where the app's windows would briefly become visible, for example as thumbnails in an app-chooser.
	Comments:
	Extracted requirement:

ID	Text
639	• The app framework should send a notification to the app program before each transition from no windows visible to one or more windows fully visible, telling it that it has been moved to the foreground (become visible).
	Comments:
	Extracted requirement:

ID	Text
642	• Until the app can redraw itself, its last known window contents must be painted. The app framework will sometimes stop apps from running, most obviously due to user request or during device shutdown. It may also stop apps if they are running in the background and there is insufficient RAM for a user-requested operation such as starting a new app, similar to the behavior of background apps in Android.
	Comments:
	Extracted requirement:

ID	Text
647	• The app framework should have a mechanism to send a request to the app process, asking it to terminate itself gracefully. (For example, systemd uses SIGTERM for the equivalent request to its managed processes.)
	Comments:
	Extracted requirement:

ID	Text
650	• A well-behaved app process should respond to this request by saving its state and terminating. The app framework must detect its termination and consider this to be a successful stop.
	Comments:
	Extracted requirement:

ID	Text

653	• The app process should update its last-used context as part of its response to this request, so that it can resume from the last-used context when started again.
	Comments:
	Extracted requirement:

ID	Text
655	• If the app process does not terminate within a reasonable time (anticipated to be limited to a few seconds), the app framework must forcibly terminate it (kill it). It must not be possible for the app process to block this forcible termination. (For example, systemd uses SIGKILL for the equivalent request to its managed processes.)
	Comments:
	Extracted requirement:

ID	Text
659	• If a stopped app is brought to the foreground, the app framework must arrange for it to be started with the last-used context.
	Comments:
	Extracted requirement:

ID	Text
661	• If the app framework needs to remove (uninstall) an app bundle that has one or more running or paused programs, it must stop those programs before commencing removal. If those programs are paused, it must unpause each one before stopping it.
	Comments:
	Extracted requirement:

ID	Text
664	If the system has a relatively large amount of RAM but a relatively slow CPU, it might be desirable to pause app processes that been sent to the background, preventing them from executing code. For example, the implementation might use SIGSTOP.
	Comments:
	Extracted requirement:

ID	Text
667	• The app framework should have a mechanism to send a request to the app process, asking it to prepare for being paused.
	Comments:
	Extracted requirement:

ID	Text
669	• The app process may respond to this request by finishing or canceling a pending operation. It should not start new operations unless they are expected to be fast.

	Comments:
	Extracted requirement:

ID	Text
671	• The app process should update its last-used context as part of its response to this request, so that if power is lost, it can resume from the last-used context when started again.
	Comments:
	Extracted requirement:

ID	Text
674	• If the app process responds to this request, it may be paused at any time after it has sent the response.
	Comments:
	Extracted requirement:

ID	Text
676	• If the app process does not respond to this request promptly (implementationdefined, but expected to be of the order of magnitude of a few seconds), it will be paused anyway.
	Comments:
	Extracted requirement:

ID	Text
679	• If the app framework notifies an app that it will be paused, but then decides that it will not actually pause the app (for example because it is brought to the foreground), it must notify the app as though it had been unpaused.
	Comments:
	Extracted requirement:

ID	Text
682	• The app must be careful to process these notifications in-order, so that if an unpause request arrives while it is still processing a pause request, the pause request is canceled. Paused apps can be unpaused, at which point they will continue to execute.
	Comments:
	Extracted requirement:

ID	Text
686	• If the app is brought to the foreground, the app framework must unpause it first.
	Comments:
	Extracted requirement:

ID	Text
687	• If a request is to be processed by the app process, for example for data sharing, document launching or URI launching, it must be unpaused first.
	Comments:
	Extracted requirement:

ID	Text
689	• If the app framework needs to stop an app program that is paused, it must unpause that app, then stop it.
	Comments:
	Extracted requirement:

ID	Text
691	• Whenever the app is unpaused, it must resume execution from the point at which it was paused, analogous to a laptop that has been placed in a "suspend to RAM" state. Shortly after it resumes execution, the app framework must either notify it that it has been unpaused, so that it can resume normal operation, or notify it that it is to be stopped, so that it can terminate itself gracefully.
	Comments:
	Extracted requirement:

ID	Text
696	• The app must be careful to process these notifications in-order, so that if an unpause request arrives while it is still processing a pause request (perhaps one for which the app framework timed out and paused it before it had responded), the pause request is canceled.
	Comments:
	Extracted requirement:

ID	Text
700	• Some design documents refer to the unpause operation as "restarting". We recommend avoiding that term, since it can mislead developers into believing that it refers to terminating the app, waiting for it to terminate, and starting it again, similar to systemctl restart.
	Comments:
	Extracted requirement:

ID	Text
704	Under some circumstances, other system components might forbid an app from being launched. For example, if an app is found to have a serious security vulnerability or contain malicious code, the system might mark it as forbidden.
	Comments:
	Extracted requirement:

ID	Text

707	• Other system components must be able to mark an installed app as forbidden. Newly forbidden apps must be stopped immediately (if running or paused), and all attempts to run them must be rejected.
	Comments:
	Extracted requirement:

ID	Text
710	• A bundle might be marked as forbidden because it contains a serious security vulnerability.
	Comments:
	Extracted requirement:

ID	Text
712	• A bundle might be marked as forbidden because it has been found to contain malicious code.
	Comments:
	Extracted requirement:

ID	Text
714	• A bundle might be marked as forbidden due to conditional access.
	Comments:
	Extracted requirement:

ID	Text
715	• Open question: is there a requirement that we can mark bundles or entry points as forbidden under specific operating conditions, for example at speeds over 20mph or at night?
	Comments:
	Extracted requirement:

ID	Text
718	• Whether a bundle is forbidden might be tracked per-user.
	Comments:
	Extracted requirement:

ID	Text
719	• A parent might use a parental control interface to mark a bundle as forbidden for their child's user account, or to limit use time so that the bundle automatically becomes forbidden after 10 minutes of use per day.
	Comments:
	Extracted requirement:

ID	Text
722	• In contexts where bundles or entry points are listed (for example by a launcher), the forbidden apps must be included in the list, with metadata indicating that they are currently unavailable. This enables vendors to make a UX decision whether to display forbidden apps (for example with a desaturated icon or a "forbidden" emblem indicating that they cannot be launched), or whether to hide them from the GUI altogether.
	Comments:
	Extracted requirement:

ID	Text
728	• The system must be able to remove the forbidden state. After this has been done, the app may be run normally.
	Comments:
	Extracted requirement:

ID	Text
730	• For example, if the app was forbidden due to a security vulnerability, the forbidden flag can be removed after upgrading it to a non-vulnerable version.
	Comments:
	Extracted requirement:

ID	Text
732	• There could be multiple reasons why an installed app is forbidden. It must be considered to be forbidden if at least one of those reasons is still valid.
	Comments:
	Extracted requirement:

ID	Text
734	• For example, if the app was forbidden due to a security vulnerability and also forbidden because its conditional-access license has expired, and an update has resolved the security vulnerability, the app must still be considered to be forbidden until a new license is obtained.
	Comments:
	Extracted requirement:

ID	Text
738	• To avoid denial of service, unprivileged apps must not be able to mark apps as forbidden.
	Comments:
	Extracted requirement:

Last-used context

ID	Text
741	The system must allow each app to store a last-used context that encodes its user-visible state during its most recent use. The last-used context must be treated as private data.
	Comments:
	Extracted requirement:

ID	Text
744	• If an app does not have any particular state, a reasonable fallback implementation is that its last-used context is the same as normal app launching. The extent to which state is saved is a quality-of-implementation issue for the individual apps: if a particular app does not save its state correctly, this is not considered a flaw in the app framework, as long as the app was given an opportunity to save its state.
	Comments:
	Extracted requirement:

ID	Text
749	• Open question: do we want to require that the app is given the opportunity to save a snapshot of its window contents, so that they can be used by the GUI shell to represent the stopped app?
	Comments:
	Extracted requirement:

ID	Text
752	If we do, then they must be stored in a prescribed location/format to be understood by the GUI shell, whereas the rest of the last-used context does not have any particular requirement about the structure or even location of the last-used context.
	Comments:
	Extracted requirement:

ID	Text
755	Alternatively, this use-case could potentially be satisfied by having the GUI shell or compositor take a snapshot without the app's involvement.
	Comments:
	Extracted requirement:

ID	Text
757	• As noted in Life-cycle management, the app program should be given the opportunity to save its last-used context before it is paused or stopped.
	Comments:
	Extracted requirement:

ID	Text

759	• The app program may save its last-used context whenever its author wishes to do so. For example, a music player might save its last-used context after it starts playing each new track.
	Comments:
	Extracted requirement:

ID	Text
762	• Long-running app programs should not save last-used context at arbitrary times (for example every 10 minutes), only when a significant event has occurred.
	Comments: JK: OK. Additionally, if an app is based on the external data/context (e.g. internet radio), it could not restore the last-used context. PW: Good point, we should add that.
	Extracted requirement:

ID	Text
764	• The app framework must be able to notify app programs that now is a good time to save last-used context.
	Comments:
	Extracted requirement:

ID	Text
766	• The app program may save its last-used context in response, but is not required to do so.
	Comments:
	Extracted requirement:

ID	Text
768	• The app program should respond to this notification. If it does not, the app framework should wait for a reasonable time (anticipated to be a few seconds) and then proceed as though it had.
	Comments:
	Extracted requirement:

ID	Text
771	• This is preferable to having long-running app programs save their state at an arbitrary time, because it gives the app framework the opportunity to influence the choice of arbitrary time. For example, the framework could notify the first app program, wait for a response, notify the second app program and so on.
	Comments:
	Extracted requirement:

ID	Text

775	• When the app is launched without any particular parameters, it must have the opportunity to load its last-used context.
	Comments:
	Extracted requirement:

ID	Text
777	• The app framework should give the app an indication of whether it is expected to load its last-used context or not.
	Comments:
	Extracted requirement:

ID	Text
779	Open question: do we expect this to be a boolean option (app should load LUC / app should not load LUC), or a tri-state (app should load LUC / app should not load LUC / app may decide)?
	Comments:
	Extracted requirement:

ID	Text
782	• Whether/when the app actually loads its last-used context is a UX decision for the platform vendor and the app vendor.
	Comments:
	Extracted requirement:

ID	Text
784	• When the app is launched for a specific purpose such as document launching or URI launching, that specific purpose takes precedence over the last-used context.
	Comments:
	Extracted requirement:

ID	Text
768	• If the app is capable of having more than one simultaneous context (for example a web browser with multiple tabs or multiple windows), the purpose for which it was launched should take precedence (for example, a tabbed web browser should load the URI from URI launching as a new foreground tab). It may additionally load its lastused context (for example, a tabbed web browser might load all the tabs from its lastused context as low-priority background tabs).
	Comments:
	Extracted requirement:

ID	Text
792	• The app framework should give the app an indication of whether, if possible, it is expected to load its last-used context in the background or not.

	Comments:
	Extracted requirement:

ID	Text
794	• Whether/when the app actually loads LUC in this case is a UX decision for the platform vendor and the app vendor. The decision made here is not necessarily the same as the decision made during launching with no particular parameters.
	Comments:
	Extracted requirement:

ID	Text
797	The app framework must also be able to store its own last-used context, consisting of the visible (foreground) app programs, and optionally some or all of the app programs that were running and/or paused in the background.
	Comments:
	Extracted requirement:

ID	Text
800	• On events such as a system reboot, the app framework may load its last-used context if desired. Whether to do this is a UX decision by the platform vendor. If it does:
	Comments:
	Extracted requirement:

ID	Text
802	• The foreground app programs should be run, each with its own last-used context.
	Comments:
	Extracted requirement:

ID	Text
803	• The background app programs may either be run with its last-used context, run with its last-used context and paused soon after, or left in the stopped state to be run with its last-used context later.
	Comments:
	Extracted requirement:

ID	Text
806	• The app framework may use the background app programs' last known window contents as a placeholder for their app window.
	Comments:
	Extracted requirement:

ID	Text

808	Open question: is this something we want? If we do, we need either a requirement that the per-app LUC includes a snapshot of the window contents in a known location/format, or a requirement that the GUI shell or compositor can take the required snapshot.
	Comments:
	Extracted requirement:

Download management

ID	Text
813	Management of app-initiated downloads has been suggested as a topic that is potentially in the scope of the app framework. We feel that this should probably be considered to be an orthogonal topic, to be designed separately.
	Comments:
	Extracted requirement:

ID	Text
816	The platform should provide a HTTP download manager for use by apps. The download manager may also be used by platform components, but that is outside the scope of a standard interface.
	Comments:
	Extracted requirement:

ID	Text
819	• It must be possible to have multiple downloads in parallel.
	Comments:
	Extracted requirement:

ID	Text
820	• The system may have a limit on the maximum number of downloads that will proceed in parallel. If it does, additional downloads must be held in a queue, with one additional download resuming every time an active download finishes successfully or unsuccessfully. This limit may be user-configurable.
	Comments:
	Extracted requirement:

ID	Text
824	• The system may start an arbitrary number of downloads in parallel, up to a specified bandwidth-usage limit. If it does, additional downloads must be held in a queue as above, with an additional download resuming when a heuristic indicates that there is enough bandwidth quota available. This limit may be user-configurable.
	Comments:
	Extracted requirement:

ID	Text

828	• Pending downloads must be saved periodically, and should be saved before system shutdown, so that they can be resumed automatically on next startup if the server supports it.
	Comments:
	Extracted requirement:

ID	Text
831	• Implementors should be aware that many servers do not support resuming HTTP downloads, either because they do not support the Range HTTP header properly or because an up-to-date session cookie is required.
	Comments:
	Extracted requirement:

ID	Text
834	• The list of pending downloads and their progress and pause/resume states must be treated as private data:
	Comments:
	Extracted requirement:

ID	Text
836	• Programs associated with an app bundle must be able to list, pause, resume and cancel the pending downloads that were started by that app bundle running as the same user.
	Comments:
	Extracted requirement:

ID	Text
839	• The progress of each pending download must be updated regularly. If a program from the initiating app is running, it must be able to receive progress reports on that download without polling.
	Comments:
	Extracted requirement:

ID	Text
842	• Programs associated with an app bundle must not be able to list, pause, resume or cancel the pending downloads that were started by a different app bundle.
	Comments:
	Extracted requirement:

ID	Text
844	• Programs running as a user must not be able to list, pause, resume or cancel the pending downloads that were started by a different user.

	Comments:
	Extracted requirement:

ID	Text
846	• The downloaded files themselves must be treated as private data:
	Comments:
	Extracted requirement:

ID	Text
847	• When an app requests that a file is downloaded, it must either be downloaded into the private data area for that (user, app) pair, or into a temporary location that is not accessible by any app. When the download is completed, if it is in a temporary location, it must be moved into the private data area for that (user, app) pair.
	Comments:
	Extracted requirement:

ID	Text
851	• It must not be possible for the app to trick the download manager into overwriting data outside its private data area, for example by creating a symbolic link and having the download manager traverse that symbolic link.
	Comments:
	Extracted requirement:

ID	Text
854	• Programs associated with an app bundle must not be able to list, pause, resume or cancel the pending downloads that were started by a (non-app) platform component.
	Comments:
	Extracted requirement:

ID	Text
856	• When a download that was initiated by an app finishes (successfully or unsuccessfully), the system must arrange for one of that app's entry points to be started (if not already running), unpaused (if paused), and notified about the status of the download. It has been suggested that the download manager should record a history of completed downloads per user, per app and/or per session.
	Comments:
	Extracted requirement:

ID	Text
862	• Open question: What are the use cases for this feature?
	Comments:
	Extracted requirement:

ID	Text
863	• If this is done, the user must be able to clear the history somehow. Without knowing the use cases for this history, we cannot say whether this should be functionality that is exposed to apps, or whether it should be considered to be a privileged action.
	Comments:
	Extracted requirement:

Installation management

ID	Text
867	Management of app bundle installation has been suggested as a topic that is potentially in the scope of the app framework. We feel that this should be considered to be an orthogonal topic, in the scope of the GENIVI Software Management design. Some requirements in this area are outlined here in the hope that they can be used to clarify the division of responsibilities.
	Comments: GN :OK
	Extracted requirement:

ID	Text
872	App bundles are expected to be user-installable, and may be updated on a schedule not matching the underlying platform.
	Comments: GN : OK
	Extracted requirement:

ID	Text
874	• Installation: New app bundles can be installed, for example from an app store.
	Comments: GN : OK
	Extracted requirement:

ID	Text
875	• It must be possible to install apps from removable storage media such as a USB thumb drive.
	Comments: GA: Optionally... I don't mind the capability being there, but not all systems will make this possible, presumably? PW: True, whether installing apps from removable media is allowed could be a vendor policy decision. Although if it’s not allowed, the system will most likely **requi re** an internet connection to install apps.
	Extracted requirement:

ID	Text
877	• Upgrade: Installed app bundles can be replaced by a newer version.
	Comments: GN: OK
	Extracted requirement:

ID	Text
878	• The system should check for upgrades periodically.
	Comments: GN : OK
	Extracted requirement:

ID	Text
879	• All programs from the app bundle must be stopped (see Life-cycle management) before proceeding with the upgrade. They must be blocked from running until the upgrade is complete.
	Comments: GN : Can this be vendor specific? In android upgrade can happen while app being run. PW: That would make things a lot more complex, in terms of guaranteeing that the upgrade happened atomically so that the old version of a running app could not accidentally load a file from the new version. As I understand it, Android does kill the application at some point during the upgrade process. In any case, the application has to be restarted at some point in order to use the upgraded binaries.
	Extracted requirement:

ID	Text
882	• If an app was installed from removable storage media, it must remain possible to upgrade it by other means (for example using an Internet connection).
	Comments: GN : OK
	Extracted requirement:

ID	Text
884	• Rollback: When an app bundle is upgraded, the version that was available prior to the upgrade must be saved, together with the state of its private data and per-app data at the time of the upgrade. The user must be able to roll back to the saved version at any time.
	Comments:
	Extracted requirement:

ID	Text
888	• Rollbacks are anticipated to be an unusual event, so the saved version may be compressed as a space/time trade-off, and its cached data may be deleted to minimize the storage cost.
	Comments:
	Extracted requirement:

ID	Text
891	• All programs from the app bundle must be stopped (see Life-cycle management) before proceeding with the rollback.
	Comments:
	Extracted requirement:

ID	Text

893	• Private and per-app data corresponding to the new version are not necessarily compatible with the saved version, so these must be rolled back too. Any changes made since the upgrade are lost.
	Comments:
	Extracted requirement:

ID	Text
896	• Removal: The user must be able to remove an installed app bundle.
	Comments: GN : OK
	Extracted requirement:

ID	Text
897	• All programs from the app bundle must be stopped (see Life-cycle management) before proceeding with the removal.
	Comments: GN : OK
	Extracted requirement:

ID	Text
899	• The app bundle's private data and per-app data must be removed. This matches what is done on Android, and is necessary to prevent a "masque attack" in which a user is induced to install a malicious bundle of the same machine-readable name from a different origin (for example via social engineering), after which the malicious bundle would be able to gain access to the private and per-app data of the original bundle.
	Comments: GN : OK
	Extracted requirement:

ID	Text
904	• Per-user data and per-device data must be unaffected.
	Comments: GN : OK
	Extracted requirement:

ID	Text
905	• Open question: it has been suggested that there should be a requirement that apps must not download in parallel, with at most one app at a time actively downloading, and the rest queued.
	Comments: GA: Who suggests it and why? blocked URL Let's discuss. I'm not sure this requirement is needed - isn't that controlled by the App Store and isn't it a (OEM) policy decision? PW: This is potentially an Apertis requirement, aimed at restricting the peak bandwidth a vehicle uses. It should probably be a vendor policy decision, yes.
	Extracted requirement:

ID	Text

908	Is this a requirement? This seems like something that should be a quality-ofimplementation decision for implementations: an implementation that expects to run on comparatively fast hardware might wish to maximize user convenience by carrying out downloads and installations in parallel, while an implementation that optimizes for implementor convenience or comparatively slow hardware might prefer to impose a limit of one download or installation at a time.
	Comments: GA: Agree, see above.
	Extracted requirement:

ID	Text
914	On a multi-user system, each user might wish to have a different set of apps installed. However, physically downloading and copying each app bundle for each user might be considered to be unacceptably inefficient.
	Comments:
	Extracted requirement:

ID	Text
917	• When a user installs an app bundle that is not yet physically installed, the system must carry out the actual installation.
	Comments: GA: I'd like to discuss this. It pertains to Software download strategy, OEM policy, and I think it could be controlled by AppStore just as well the embedded system? PW: It pertains to the software download strategy, and the way that user accounts are separated and where apps are installed — if apps are installed in any kind of system-wide prefix, then the converse of this requirement is hard to meet. GA: I'd be just as happy leaving out this requirement - it adds little understanding in my opinion. PW: OK. It's basically determined by whether the bundles are installed system-wide or not.
	Extracted requirement:

ID	Text
919	• When a different user is active, the system should behave as if that app bundle was not physically installed: it must not be run, its entry points must not be available for launching or data sharing, and so on.
	Comments: GA: Policy decision? But yes, agree the capability must be there. PW: This could be a policy decision — the choice here basically depends on whether the vendor has chosen for users to have strong privacy from other users. If they have weak privacy, it would make more sense for all installed apps to be listed in each user's launcher.
	Extracted requirement:

ID	Text
922	• As an exception to that general rule, privileged app management GUIs should be able to enumerate the app bundles that are physically installed, for example so that they can illustrate how storage space has been used.
	Comments: GA: OK
	Extracted requirement:

ID	Text
925	• This could usefully be implemented by treating it as forbidden for the other users.
	Comments: GA: OK
	Extracted requirement:

ID	Text
926	• When a user installs an app bundle that has already been physically installed by another user, the system must stop hiding the app bundle from that user. For example, it must now be made available for launching by that user, assuming there is no other reason why it would be forbidden.
	Comments: GA: For me this is implicit from the previous "store (code) only once" requirement (if we keep that one) PW: We wanted to make the user-visible effects of another user 'installing' the app explicit. GA: I understand but I think at the high evel" of requirements we now have, whether apps are "unhidden" or installed anew is an implementation detail. GA: I could live without this text as a requirement. PW: OK. I think the important requirement overall in this section is that if user A installs an app, user B (or their apps) must not be able to detect the app has been installed until they decide to install it themselves.
	Extracted requirement:

ID	Text
930	• If a user has installed an app from a particular origin, then another user is not required to be able to install an app of the same name from a different origin.
	Comments: GA: Hmm. Same name means same identity? Let's dig into this a bit... PW: 'Same name' means same identifier, yes. Identifiers for apps are meant to be globally unique; we have been using a reverse-DNS notation for this, for example 'org.example.MyCalendarApp'. GA: Let's change name to "application identifier" or similar - and put whatever we choose into the definitions table. PW: OK GA: But I think it could be solved by simply stating that application identifiers are unique, whether this is possible to enforce or not, it ought to be the fundamental idea, right? The requirement here sounds like a description of a strategy to handle an exceptional event, i.e. in case we ever encounter two apps with identical identifier, but it does not seem to cover every error case anyway: "another user" - what if the *same user* is requesting the second installation? Etcetera. PW: This section exists in response to the iOS masque attack. I guess the basic requirement is that application identifiers are globally unique, across all installation origins.
	Extracted requirement:

ID	Text
932	• If a user has installed an app at a particular version, then another user is not required to be able to install a different version of that app.
	Comments: GA: Agree but it should be simple to understand. Basically just guarantee all users have the same version. (follows from "store only once" idea). PW: I think the wording is this way round so that it's not *disallowed* for users to be running different versions of the same app — some implementations might allow this, and might strive for it in order to implement strong privacy between users. Apertis does not allow this: users must run the same version of each app.
	Extracted requirement:

ID	Text
934	• If a user upgrades or rolls back an app, the app may be upgraded or rolled back for all other users.

	Comments: GA: Agree. Basically I a simple model is desired. Are users at all involved in which version of an app is being executed? PW: Do you mean in terms of being prompted about upgrades? I was assuming we would go with an Android-style model where the user is told which apps are going to be upgraded soon, and then the upgrade happens automatically unless the new version of an app requires the user to give it more permissions. That would require user intervention. However, as far as I am aware, no design work has happened on the user interaction for this yet. GA: This strategy I think is again OEM policy... So the possibility of doing this should be there, but it's not the only way. PW: Yes. This bullet point exists because it affects where and how apps are installed: whether the system needs to keep multiple versions of a single app around for different users. So the choice here significantly affects the application framework implementation, but shouldn't affect the design too much.
	Extracted requirement:

ID	Text
936	• Open question: do we want to mandate that the physical installation of apps must be per-device, or leave that open?
	Comments: GA: Don't get it, please explain. AppStore/policy decision not affecting the device right? Or do you mean "store only once" requirement. PW: This is the 'store only once requirement' — installing apps per-device as opposed to per-(user, device). GA: OK, first of I would then rewrite "per-device" into something better. GA: Then I'm not sure. I waiver between thinking it is reasonable to assume every code is only stored once, vs. this being an implementation decision. If you have enough memory, and storing all applications inside a filesystem namespace that is unique to each user might still be a simple and effective option? Maybe even to the point of bundling all dependencies in app bundle... PW: I think this could indeed be an OEM decision. It significantly affects the implementation of an application framework, but shouldn't affect the design much.
	Extracted requirement:

ID	Text
938	A vendor might wish to include app bundles in the original factory state of the system, while subsequently allowing them to be upgraded and uninstalled by the user, in the same way that Google apps are typically handled on Android devices.
	Comments:
	Extracted requirement:

ID	Text
941	• Preinstalled apps: it must be possible to preinstall app bundles on the system, while leaving them available for installation management (upgrade, rollback, removal) in the usual way.
	Comments:
	Extracted requirement:

Conditional access

ID	Text
945	App-store curators and app vendors might wish to provide publish apps on a time-limited basis. This is a complex topic and we recommend that it is considered separately. The Apertis Conditional Access design has some proposed requirements for this topic.

	Comments: GN : OK GA: I agree. Isn't this simply covered by some general mechanism in which OEMs can forcibly remove apps from the installation (security problem, time limited, or deprecated for **any** reason). Then it becomes OEM policy decision what to do with that possibility. Let's build in the requirements of keeping track of installation time and other such mechanisms (must be secure and not subvertible). PW: Agreed. For the Apertis conditional access design, we need: timestamp of installation or upgrade of a bundle; signature of bundle integrity from the app store; globally unique user, device, vehicle and bundle identifiers (note that there might be multiple Apertis devices in a single vehicle, and licensing could be separate for all of them); a way to work out when a trip ends in a vehicle (so it doesn't remove access to an app part-way through a trip).
	Extracted requirement:

Appendix: mapping to GENIVI Platform Compliance Specification 10.0

ID	Text
951	• SW-APPFW-AM-001 Manifest file for Application: this is the bundle metadata, the app permissions, and the entry point metadata (including the details demanded by document launching and URI launching). Open question: Do we need an explicit statement of what else would go in here, like required API levels?
	Comments: GN : This review comment is considered and updated the Requirement before the SAT approval for Miranda Compliance.
	Extracted requirement:

ID	Text
956	This appears to be taking an implementation detail (the manifest file) of the motivating requirements (framework must be able to []) and declaring it to be a requirement in its own right. We have attempted to re-state it in terms of requirements.
	Comments: GN : This review comment is considered and updated the Requirement before the SAT approval for Miranda Compliance.
	Extracted requirement:

ID	Text
960	• SW-APPFW-AM-002 Support for LUC: Last-used context
	Comments: GN : OK
	Extracted requirement:

ID	Text
961	• SW-APPFW-AM-003 Failure handling in case of application doesn't respond on state change: Life-cycle management
	Comments: GN : OK
	Extracted requirement:

ID	Text
963	• SW-APPFW-AM-004 Launch application from another application: this is document launching, URI launching and perhaps app launching.
	Comments: GN : OK
	Extracted requirement:

ID	Text
965	• SW-APPFW-AM-005 Factory reset: Data management
	Comments: GN : OK
	Extracted requirement:

ID	Text
966	• SW-APPFW-AM-006 Prohibit to start an application: see Life-cycle management and specifically Forbidden apps.
	Comments: GN : OK
	Extracted requirement:

ID	Text
968	• SW-APPFW-AM-007 Activation of application, SW-APPFW-AM-008 Deactivation of application: What is activation?
	Comments: GN : This review comment is considered and updated the Requirement before the SAT approval for Miranda Compliance.
	Extracted requirement:

ID	Text
970	• SW-APPFW-AM-009 Support for activation of application (sic): from its descriptive text, this seems to actually be app launching.
	Comments: GN : This review comment is considered and updated the Requirement before the SAT approval for Miranda Compliance.
	Extracted requirement:

ID	Text
972	• SW-APPFW-AM-010 Support for switching the application (sic): from its descriptive text, this seems to actually mean stopping the application. Life-cycle management
	Comments: GN : This review comment is considered and updated the Requirement before the SAT approval for Miranda Compliance.
	Extracted requirement:

ID	Text
975	• SW-APPFW-AM-011 Support for pausing an application: Life-cycle management
	Comments:
	Extracted requirement:

ID	Text
976	• SW-APPFW-AM-012 Support for resuming application: Life-cycle management

	Comments: GN : OK
	Extracted requirement:

ID	Text
977	• SW-APPFW-AM-013 Support for stopping application: from its descriptive text, this is specifically stopping a paused application. Life-cycle management
	Comments: GN : This review comment is considered and updated the Requirement before the SAT approval for Miranda Compliance.
	Extracted requirement:

ID	Text
979	• SW-APPFW-AM-014 Application framework shall provide a mechanism to tell an application to change its state: the states specified are START (not running), BACKGROUND (running and in background), SHOW (running and in foreground), RESTART (from its descriptive state not actually a state, and not the systemd-style restart action either, but in fact the "resume" transition from PAUSE to either SHOW or BACKGROUND), OFF (what is the difference between this and START in terms of states?), and PAUSE (understood to be essentially SIGSTOP'ed). See Life-cycle management.
	Comments: GN : This being worked out.
	Extracted requirement:

ID	Text
987	These state names demonstrate some confusion between states and state transitions. We have specifically documented states, not transitions, and provided details of the allowed transitions.
	Comments: GN : This being worked out
	Extracted requirement:

ID	Text
990	• SW-APPFW-AM-015 Application states: the states specified are either (INSTALLED, ACTIVATED, LAUNCHED, PAUSED) or (START, BACKGROUND, SHOW, RESTART, OFF, PAUSE) depending which column we believe. See Lifecycle management. It is unclear what these states mean, particularly ACTIVATED. We have described a different set of states in these requirements.
	Comments: GN : This being worked out
	Extracted requirement:

ID	Text
996	• SW-APPFW-AM-016 Installed application info: this is the part of app launching that deals with listing what we can launch.
	Comments: GN : OK
	Extracted requirement:

ID	Text

998	• SW-APPFW-AM-017 Access restriction for apps: this is our sandboxing and security. It's a big topic in its own right.
	Comments: GN : Agreed. This is discussed under access mechanism.
	Extracted requirement:

ID	Text
1000	• SW-APPFW-AM-018 Support for different applications running in different runtimes: the application framework should support JVM- or HTML5-based runtimes. Stated in What's in an app.
	Comments: GN : This review comment is considered and updated the Requirement before the SAT approval for Miranda Compliance.
	Extracted requirement:

ID	Text
1003	• SW-APPFW-AM-019 Support for any number of applications: stated in What's in an app, under the assumption that this is referring to lack of arbitrary limits. If the intention is to cope with exceeding RAM by telling excess apps to shut down gracefully, that's harder but could be done. If the intention is to cope with exceeding flash space by "swapping out" apps to cloud storage or something, that's impractical for a device that might not have constant connectivity and should not be required.
	Comments: GN : This review comment is considered and updated the Requirement before the SAT approval for Miranda Compliance.
	Extracted requirement:

Appendix: mapping to Suma's proposed requirements

ID	Text
1011	• App-FW-001 Protect the system against altering of any data by a malicious app: App integrity, System integrity, Per-user data, etc.
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1013	• App-FW-002 Protect the system against collecting and sharing of any data by a malicious app: App confidentiality, Private data, Per-user data etc.
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1015	• App-FW-003 Protect the system against usage of system resources etc.: Resource limits
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1017	• App-FW-004 An application shall not [~gunnar.andersson:] interfere with [~gunnar.andersson:] the … … other application: App integrity, App confidentiality, Private data
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1019	• App-FW-005 read, alter or delete non-application data: System integrity, Peruser data.
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1021	As written, this requirement states that this must be forbidden entirely. We have assumed that the intention was to forbid it with exceptions where necessary for the app to do its job.
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1024	• App-FW-006 Users data are protected against access by another user: Private data, Per-user data
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1026	• App-FW-007 deny access to APIs to which an App has not requested permission: Sandboxing and security
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1028	This requirement wrongly conflates APIs with privilege boundaries. There is never any reason to deny access to APIs that do not cross a privilege boundary, because such APIs cannot do anything that the app could not do itself.
	Comments: GN : Agreed.
	Extracted requirement:

ID	Text

1032	• App-FW-008 per-app rollback: Rollback
	Comments: GN : under discussion
	Extracted requirement:

ID	Text
1033	• App-FW-009 Shall support applications with UI or UI less: What's in an app
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1034	• App-FW-010 Restore LUC: Last-used context
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1035	• App-FW-011 information about mime type: Document launching Consideration has been given to possible ways to select file types, other than media types. We have included the recommendation that using anything other than IETF media types would be unwise.
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1036	• App-FW-012 Resource handling: Life-cycle management
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1040	• App-FW-013 Inform apps about states: Life-cycle management
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1041	• App-FW-014 shutdown: Life-cycle management
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1042	• App-FW-015 Frozen state: Life-cycle management (we're calling it "pause" in this document)
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1044	• App-FW-016 blacklist apps: We think this may be conflating two distinct behaviors. The first is to cope with apps that go into a crash loop, which must be rate-limited. The second is to have a way to stop apps executing altogether, which this document refers to as Forbidden apps.
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1049	• App-FW-017 apps with a validity period: Conditional access
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1050	• App-FW-018 app requesting permissions every launch: App permissions. Note that we only really recommend this for permissions where there's nothing better we can do, like "unrestricted Internet access".
	Comments:
	Extracted requirement:

ID	Text
1053	• App-FW-019 apps can communicate with other apps: Data sharing
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1054	• App-FW-020 Content hand-over: Document launching, URI launching.
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1055	• App-FW-021 content type can be opened only by...: Document launching

	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1056	• App-FW-022 It shall be possible for an app to register a new content type: Adding media types
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1058	• App-FW-023 Sharing a content to be transferred out of the system: (Androidstyle Sharing API): Sharing menu
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1060	• App-FW-024 POI provider but no access to location data: implicit in sandboxing and security and app permissions. This requirement appears to be conjecturing that registering an app as a points-of-interest provider would cause it to have additional permissions somehow, but whether an app is registered as a points-of-interest provider should be entirely orthogonal to whether it has the permissions that would allow it to access location data.
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1067	• App-FW-025 to App-FW-032 Download manager: Download management
	Comments:
	Extracted requirement:

ID	Text
1068	• App-FW-032 to App-FW-036 Internationalization: not mentioned here. As Gunnar says, this is a SDK API issue, not a platform services issue. It is entirely feasible to implement internationalization through a shared library provided by the platform (part of glibc in practice) and some data files in the app (gettext .mo files) without ever crossing a security boundary, and we recommend doing exactly that.
	Comments: GN : Agreed
	Extracted requirement:

ID	Text
1074	• App-FW-037 installation of application bundles: Installation

	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1075	• App-FW-038 Native application: we are unsure how this is relevant to a
	Comments: GN : Not in the scope of Managed Apps handling.
	Extracted requirement:

ID	Text
1076	GENIVI design, since the interaction between vendor-supplied native apps and the vendor-supplied platform is presumably up to the vendor.
	Comments: GN : We need to cover those areas which are coomon and can be generalized.
	Extracted requirement:

ID	Text
1078	Terminology note: GENIVI's native applications are the same thing as Apertis' built-in applications. It is nothing to do with whether the app is written in native code compiled from C/C++. GENIVI applications that are not native applications are said to be managed applications, which are the same as Apertis' store applications.
	Comments:
	Extracted requirement:

ID	Text
1083	• App-FW-039 Pre installed app vs. store downloadable apps: Preinstalled apps
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1084	• App-FW-040a Install app from a storage device: Installation
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1085	• App-FW-040b sync up with app store: We have interpreted this to mean that after installation from removable media, it must still be possible to upgrade via the Internet.
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text

1088	• App-FW-041 facilitate handling of permissions: app permissions
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1089	• App-FW-042 provide data storage structure to an app: private data and optionally per-app data, per-device data, per-user data.
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1091	• App-FW-043 an app can't contain more than one program … or more than one agent/service: What's in an app
	Comments:
	Extracted requirement:

ID	Text
1093	There has been some resistance to this requirement, and we have written the requirements in this document to say that vendors may impose this limit, but the framework should not.
	Comments: GN : Agreed.
	Extracted requirement:

ID	Text
1096	• App-FW-044 system extensions: What's in an app
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1097	• App-FW-045 downloaded and installed only once (i.e. apps appear to be peruser but are really system-wide): Installation management
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1099	• App-FW-046 queueing mechanism for app download (i.e. apps do not install in parallel): Software download limiting
	Comments: GN: OK, type : Req mapping
	Extracted requirement:

ID	Text
1101	• App-FW-047 App upgrades shall be checked periodically: Upgrade.
	Comments: GN: OK, type : Req mapping
	Extracted requirement: