

Proposal for Scalable Components

Document WINNF-14-R-0010

Version V1.0.0

25 July 2014

Terms and Conditions

This document has been prepared by the SCAv4.1 Backwards Compatibility Task Group to assist The Software Defined Radio Forum Inc. (or its successors or assigns, hereafter “the Forum”). It may be amended or withdrawn at a later time and it is not binding on any member of the Forum or of the SCAv4.1 Backwards Compatibility Task Group.

Contributors to this document that have submitted copyrighted materials (the Submission) to the Forum for use in this document retain copyright ownership of their original work, while at the same time granting the Forum a non-exclusive, irrevocable, worldwide, perpetual, royalty-free license under the Submitter’s copyrights in the Submission to reproduce, distribute, publish, display, perform, and create derivative works of the Submission based on that original work for the purpose of developing this document under the Forum's own copyright.

Permission is granted to the Forum’s participants to copy any portion of this document for legitimate purposes of the Forum. Copying for monetary gain or for other non-Forum related purposes is prohibited.

Intellectual Property Rights

THIS DOCUMENT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE FORUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS DOCUMENT.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

Proposal

This document contains a proposal to change the SCAv4.0.1 specification to add support for scalable components. This proposal will allow SCAv4.1 components to selectively implement different standardized interfaces.

Proposal author:

- Gerald L Bickle, Raytheon*

Proposal reviewers:

- Steve Bernier, NordiaSoft
- François Lévesque, NordiaSoft
- Hugues Latour, Communications Research Centre Canada
- David Hagood, Aeroflex
- Eric Christensen, JTNC

*Special thanks goes to Gerald Bickle for having prepared the bulk of the material including all the annotated specification documents part of this proposal.

Recommendation

Component Scalable UOF Changes

Topics

Scalable Component Approach

ComponentBase

ComponentFactoryComponent

Application Component Changes

Device Component Changes

Platform Component Changes

Appendix F Changes

Update Appendix C based upon Spec Changes

Scalable Component Approach

Approach should be UML compliant as much as possible for modeling Application, Device, and Service components.

- Removes optional IDL inheritance for Resource and Device interfaces.
- User Defined interfaces for scalable Resource, Device, and Service interface types (e.g., MyResource, MyDevice, MyService).

Allow for full and scalable Application, Device and Service components.

Remove Resource and Device interfaces.

- Backwards Application Capability appears to be removing SCA V4 Resource Interface
- These Interfaces are not reference by Component definitions in scalable approach

Figure 2-3: SCA Component Hierarchy needs to be updated base upon changes in this presentation

Figure 2-5 Conceptual Model of Resources needs to remove or updated

Scalable Component Approach

Figure 3-2: Core Framework IDL Relationships
needs to be by removing interfaces and changing
any names, changing any relationships

**CF Manager Components could follow similar
Scalable Approach**

**Component and Interface Names may be changed
later based upon Naming convention resolutions.**

ComponentBase

Change Figure 3-5

- ComponentBase UML realized relationships to an aggregation and label the aggregation the same name.
- Add association to ComponentRegistry
- Removed UtilityComponent

Modified 3.1.3.1.2.1.2 associations

- Removed UtilityComponent
- Added registrar: A ComponentBase may register its reference using the ComponentRegistry interface.

3.1.3.1.2.1.3 Semantics

- Added
 - In addition to supporting the CF Base Application interfaces, a ComponentBase may implement and use component-specific interfaces for data and/or control. Interfaces provided by a ComponentBase are described in a Software Component Descriptor (SCD) file as provides ports. Interfaces required by a ComponentBase are described in a SCD file as uses ports.

ComponentBase, cont'd

3.1.3.1.2.1.4 Constraints

- Modify Requirements
 - “SCA427 A ComponentBase shall be associated with an SPD file” to “SCA427 A ComponentBase shall be associated with a domain profile file”
- Add Requirements
 - A ComponentBase may realize the *PropertySet* interface to configure and query its configure and query properties.
 - A ComponentBase may realize the *TestableObject* interface for its test properties.
 - A ComponentBase may realize the *PortAccessor* interface for its uses and provides ports.
 - A ComponentBase may register its component reference via the *ComponentRegistry* interface.
 - SCA_TBD The component reference shall consist of its base interfaces composition.

ComponentBase, cont'd

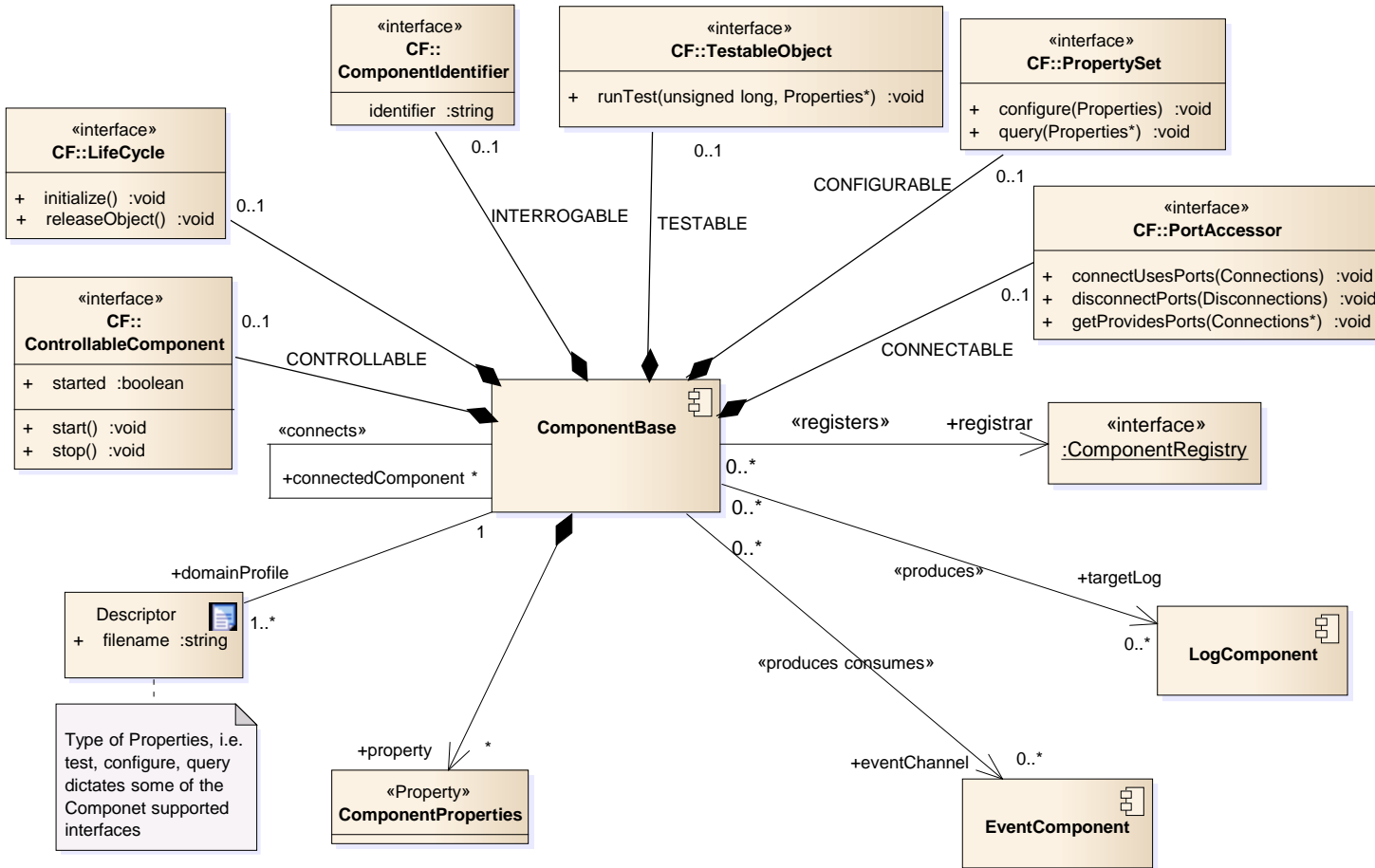
3.1.3.1.2.1.4 Constraints (...)

- Remove Requirements
 - “SCA426 A ComponentBase shall realize the ComponentIdentifier “ to “SCA426 A ComponentBase may realize the ComponentIdentifier “
 - “SCA432 A ComponentBase shall realize the *LifeCycle* interface.” to “SCA432-A ComponentBase may realize the *LifeCycle* interface. A ComponentBase that is initializable via the *LifeCycle* interface may not be releasable.”
 - SCA433 A ComponentBase shall realize the *ControllableComponent* interface to provide overall management control of the component to “A ComponentBase may realize the *ControllableComponent* interface to provide overall management control of the component”

Appendix F add mapping to new optional requirements

- LifeCycle ReleaseObject. – Releasable UOF

Component Base, cont'd



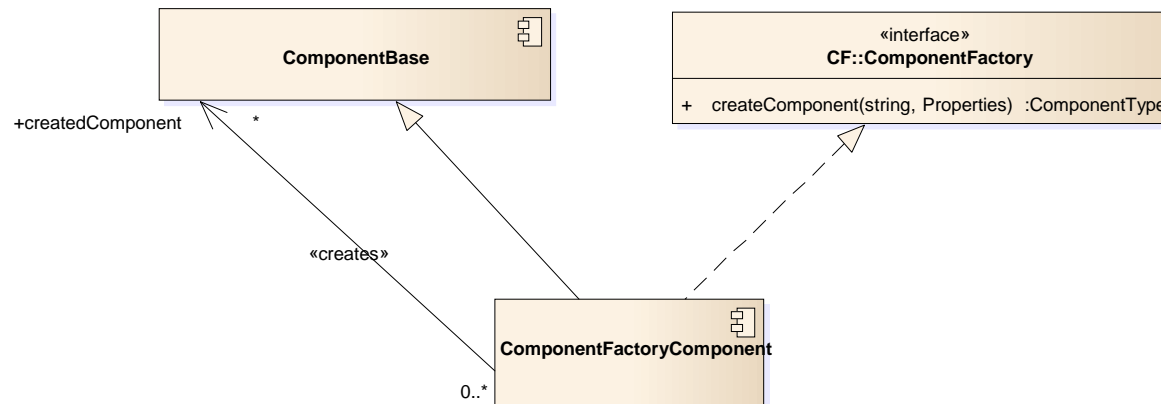
ComponentFactoryComponent

Update Figure 3-6 ComponentFactoryComponent UML by removing associations that are already covered by ComponentBase and also ServiceComponent is a type of ComponentBase now.

Update 3.1.3.1.2.2.2 Associates – removing createdService and ComponentRegistry

Add constraint, 3.1.3.1.2.2.4

- SCA_TBD A ComponentFactoryComponent shall realize the *LifeCycle* interface.



Application Components Changes

Remove Resource Interface – Backwards Application Capability is removing this

Remove ResourceComponent since application components may not be a full Resource.

Change Name of Application Resource Component to CF_ApplicationComponent

Modify

- Application Component
- Assembly Controller Component
- ApplicationComponentFactoryComponent

Application Components Changes

In Application Components Section change order of components to be

- ApplicationComponent
- CF_ApplicationComponent
- AssemblyControllerComponent
- ApplicationComponentFactoryComponent

ApplicationComponent

Description

- “An ApplicationComponent is ~~an abstract~~ a component that captures the common requirements ~~of the for~~ SCA ~~Base~~ Application Components. ~~An ApplicationComponent is a constituent part of an AssemblyComponent.~~”
- No longer abstract since definition is for any application component type (CF or non-CF)

Update Figure

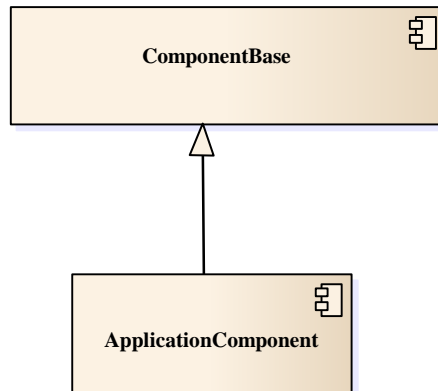
Semantics

- Move and modify SCA82 registration requirement to new CF_ApplicationComponent.

Constraints

- Move SCA 166 and SCA 167 to new CF_ApplicationComponent.
- Reword SCA 169 by replacing “the appropriate Domain Profile Files” with “Software Package Descriptor Domain Profile File”
- Added
 - SCA_TBD An ApplicationComponent shall fulfill the ComponentBase requirements.

ApplicationComponent, cont'd



CF Application Component

Change CFApplicationResourceComponent name to CF_ApplicationComponent throughout section.

- Need a different name since this may not be a full Resource interface.
- Following CF_Service naming convention.

Description

- Change - ~~An ApplicationResourceComponent is a constituent part of an AssemblyComponent. An CF_ApplicationResourceComponent, a specialization of ResourceComponentApplicationComponent, provides a common API for control and configuration of ResourceComponent application component within the context of a deployed application.~~

Update Figure

3.1.3.2.2.2 Associations

- Remove domainProfile covered by ComponentBase
- Add proxyComponent: An CF_ApplicationComponent may interface with its AssemblyControllerComponent.

CF Application Component

3.1.3.2.2.4.3 Semantics

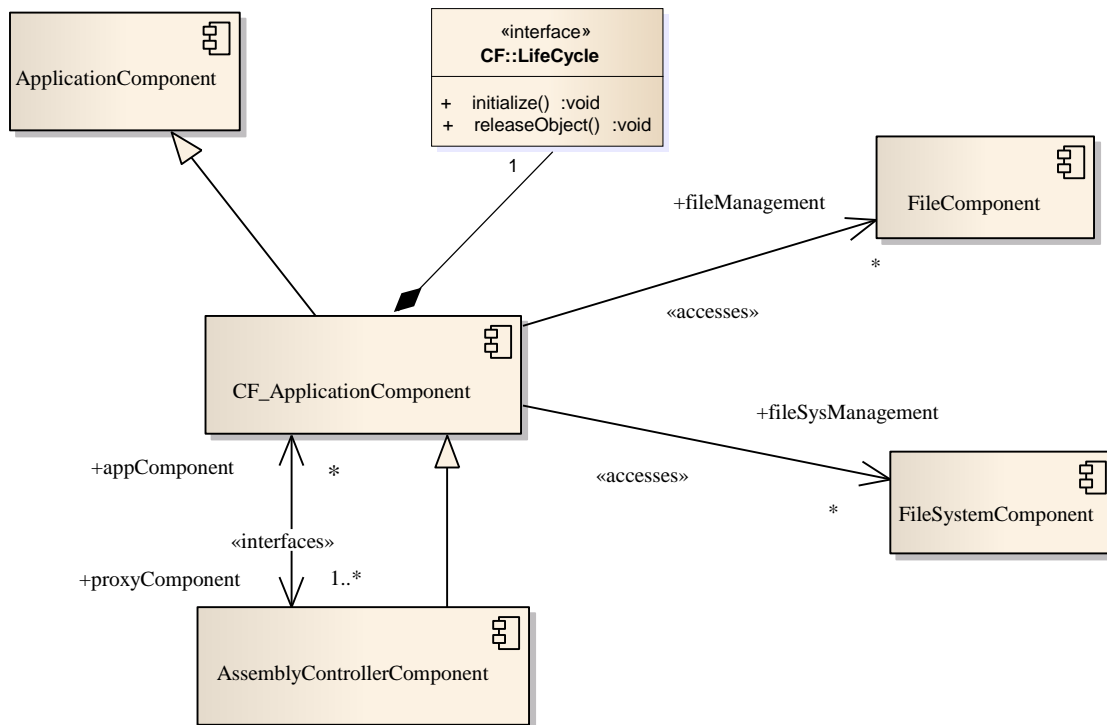
- Add and modify SCA82 registration requirement – from “SCA82 An ApplicationComponent created via an ExecutableDeviceComponent shall register with its launching ApplicationFactoryComponent via the ComponentRegistry::registerComponent operation using the Component Registry execute parameter.”
- Remove Statement since covered in ComponentBase – “In addition to supporting the CF Base Application interfaces, an ApplicationResourceComponent may implement and use component-specific interfaces for data and/or control. Interfaces provided by an ApplicationResourceComponent are described in a SCD file as provides ports. Interfaces required by an ApplicationResourceComponent are described in an SCD file as uses ports.”

CF Application Component

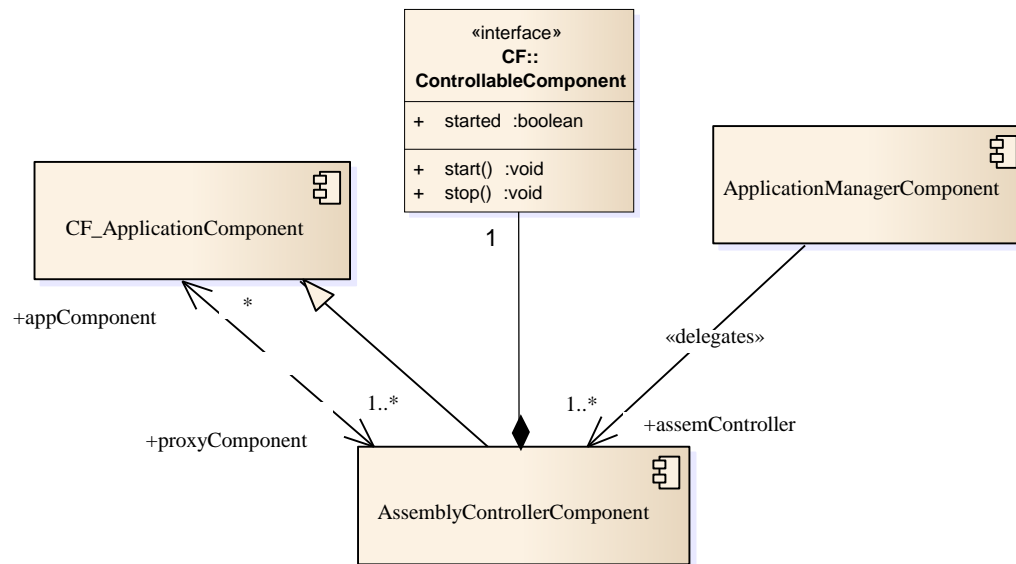
3.1.3.2.2.4.4 Constraints

- Remove
 - SCA172 Req sine no longer this type – “An ApplicationResourceComponent shall fulfill the ResourceComponent requirements”
 - Removed “Dynamically created stringified IORs...” statement
- Add
 - SCA 166 and 167 file system requirements from ApplicationComponent
 - SCA_TBD A CF_ApplicationComponent shall realize the LifeCycle interface.
 - Add “SCA_TBD A CF_ApplicationComponent shall set its registeringComponent type to be APPLICATION_COMPONENT.
 - Tied to Optional Component Registration UOF

CF_ApplicationComponent, cont'd



Changed “ApplicationComponent” to “CF_ApplicationComponent” throughout section

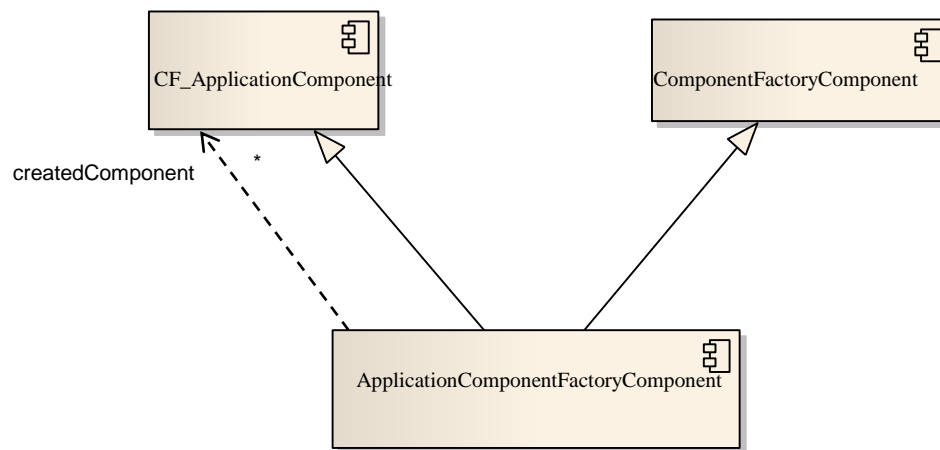


ApplicationComponent

FactoryComponent

Changed “ApplicationComponent” to
“CF_ApplicationComponent” throughout section.

Updated Figure



Device Changes

3.1.3.4.1 Interfaces

- Change “The device interfaces are Device, LoadableDevice and ExecutableDevice.” to “The device interfaces are AggregateDevice, CapacityManagement, DeviceAttributes, LoadableDevice ManageableComponent, ParentDevice, LoadableObject and ExecutableDevice.”

Remove Device Interface

- Move Device Release Object Behavior and Exceptions text to DeviceComponent section.

Remove LoadableDevice or LoadableDevice becomes LoadableObject

- This depends on what happens with Backwards Compatible
- LoadableObject Interface
 - Change Description from “This interface extends the LoadableDevice interface by adding software loading and unloading behavior to a device” to “This interface provides software loading and unloading behavior for a device”

ExecutableDevice remove the inheritance

- Depending on what happens with Backwards Compatible name may be changed to ExecutableObject
- Description is only first sentence. “This interface provides execute and terminate behavior for a device”
- Update Figure
- Remove section 3.1.3.4.1.8.5.3 releaseObject

Device Changes

Change Component Base Device to be Device Component and remove existing Device Component

Remove Old Device Component

Modify LoadableDevice Component

Modify ExecutableDevice Component

Modify AggregateDevice Component

ExecutableObject Interface Change

«interface» ExecutableObject
+ PRIORITY_ID :string = "PRIORITY" {readOnly} + STACK_SIZE_ID :string = "STACK_SIZE" {readOnly}
+ execute(string, Properties, Properties) :ProcessID_Type + terminate(ProcessID_Type) :void

Device Component

In section 3.1.3.4.2.1 Change Component Base Device to be Device Component throughout the section since both are not needed in Scalable Approach changes

3.1.3.4.2.1.1 Description

- Remove sentence “This abstraction is necessary because even though the corresponding interfaces for those components do not share an inheritance relationship among themselves they share a common collection of interfaces. This secondary relationship allows the components to have the same baseline capabilities.”
- Change sentence - A ~~DeviceComponentBaseDevice~~ is an abstract component that extends PlatformComponentBase. ~~ComponentBaseDeviceComponent~~ contains the core associations and requirements that are used by the SCA device oriented components (~~DeviceComponent, LoadableDeviceComponent and ExecutableDeviceComponent~~).

Device Component

- Change Figure 3-53 DeviceComponent UML optional realized relationships to an aggregation and label the aggregation the same name.

Update 3.1.3.4.2.1.2 associations

- Remove componentRegistry since covered by ComponentBase

DeviceComponent, cont'd

3.1.3.4.2.1.3 Semantics

- Move and modify “ SCA298 A ComponentBaseDevice shall register with its launching DeviceManagerComponent via the *ComponentRegistry::registerComponent* operation” to PlatformComponent
- Add “SCA_TBD A DeviceComponent shall set its registeringComponent type to be DEVICE_COMPONENT.”
 - Tied to Optional Component Registration UOF
- Move 3.1.3.4.1.1.5.1.3 Section text to here
 - Tied to optional Releasable UOF
- Move 3.1.3.4.1.1.5.1.5 Section text to here
 - Tied to optional Releasable UOF

DeviceComponent, cont'd

3.1.3.4.2.1.4 Constraints

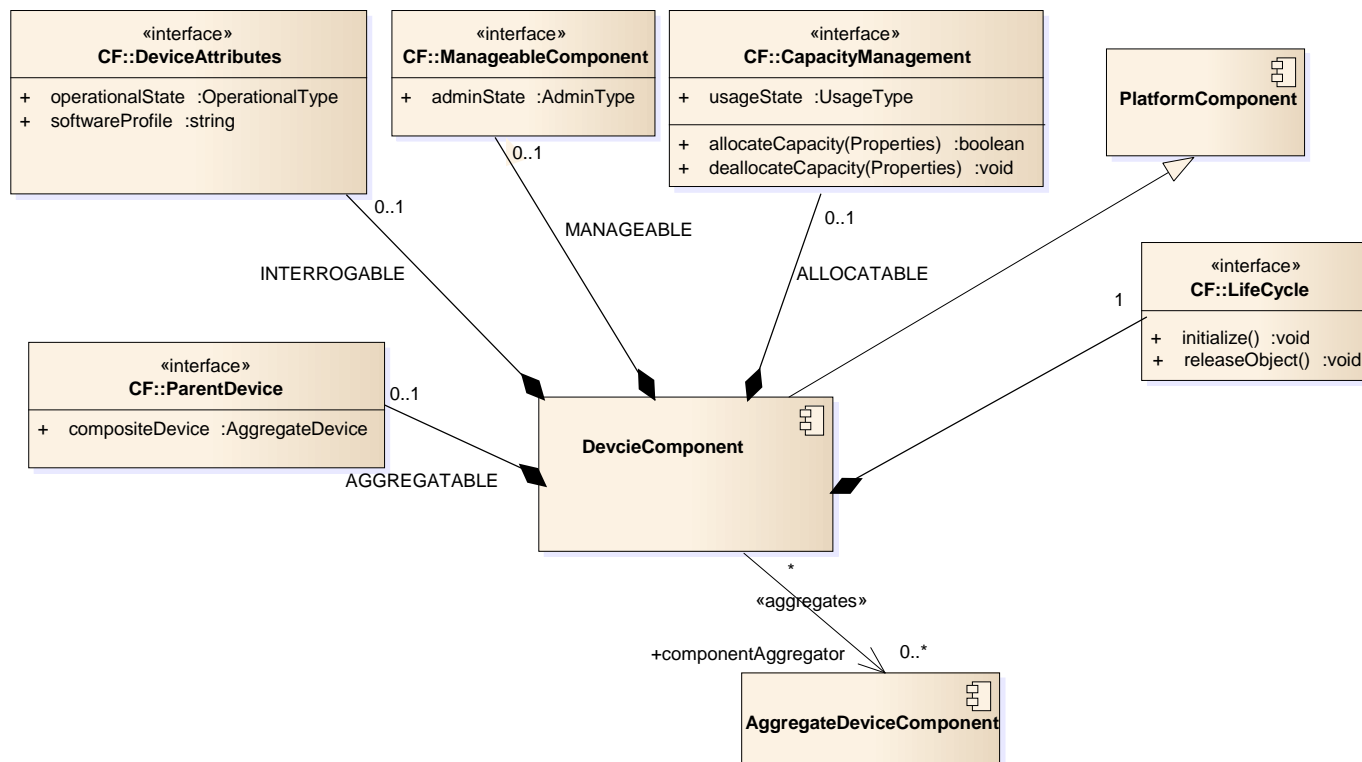
- Remove requirement since covered at platform component level.
 - SCA303 A ComponentBaseDevice shall fulfill the ComponentBase requirements
- Add Req
- SCA_TBD A DeviceComponent shall realize the *LifeCycle* interface.
- Modify Reqs since optional
 - ~~SCA534~~ A ComponentBaseDeviceComponent shall may realize the DeviceAttributes interface. ~~SCA535~~ A ComponentBaseDeviceComponent shall may realize the ManageableComponent interface. ~~SCA536~~ A ComponentBaseDeviceComponent shall may realize the CapacityManagement interface. ~~SCA539~~ A ComponentBaseDeviceComponent shall may realize the ParentDevice interface”

Appendix F

- Remove Device Releasable since Releasable is now at ComponentBase.

Remove old 3.1.3.4.2.2 Device Component Section

Device Component, cont'd



Loadable Device Component

Change Description

- The LoadableDeviceComponent extends the ~~ComponentBaseDevice-component~~ DeviceComponent by adding software loading and unloading behavior

Update Figure

- Change interface from LoadableDevice to LoadableObject
- Change inheritance to DeviceComponent

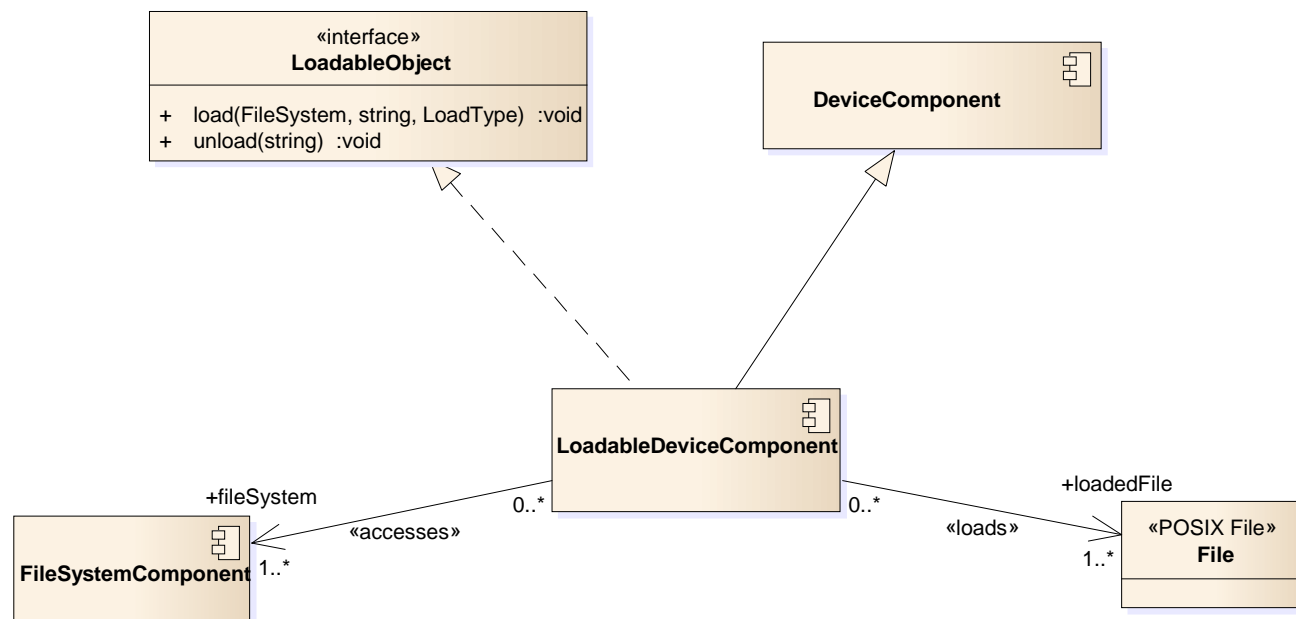
3.1.3.4.2.3.3.1 State Model

- Replace in text “device’s” with “LoadableDeviceComponent’s”

Change requirements

- A LoadableDeviceComponent shall realize the *Loadable* ~~ObjectDevice~~ interface.
- SCA309 A LoadableDeviceComponent shall fulfill the ~~ComponentBaseDeviceComponent~~ requirements.

LoadableDeviceComponent Figure



Executable Device Component

In section replace “ComponentBaseDevice” with
“DeviceComponent”

Update Figure

Associations

- “executableArtifact: An ExecutableDeviceComponent loads, unloads, executes and terminates artifact(s) (i.e. processes, executables or modules) within a processing environment

3.1.3.4.2.4.3.1 State Model

- Replace in text “device’s” with
“ExecutableDeviceComponent’s”

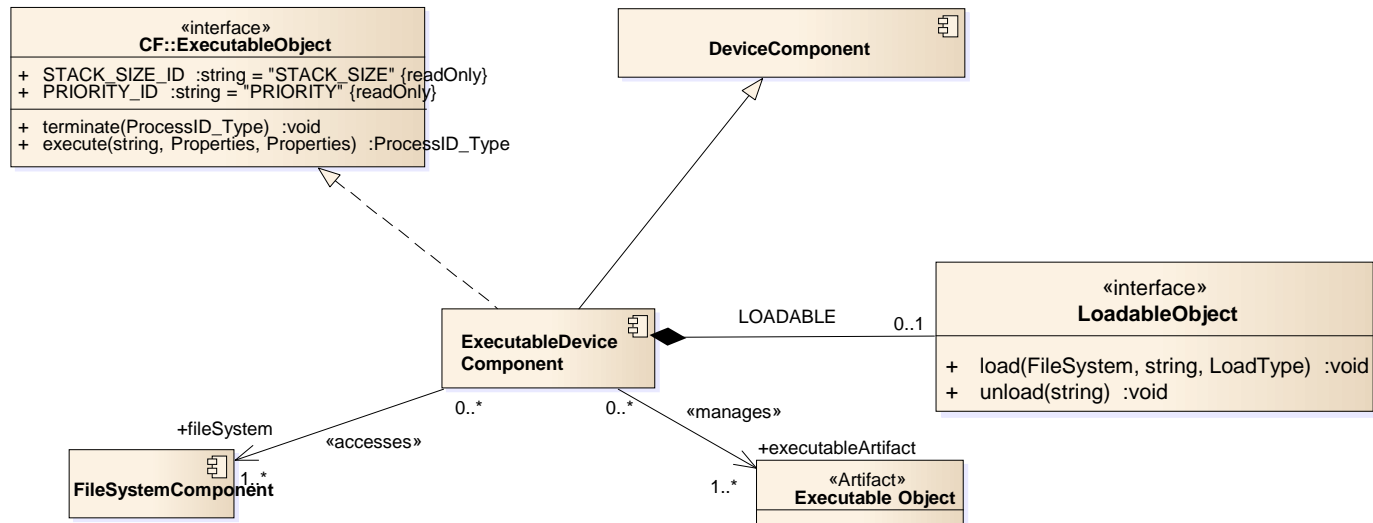
Added statement

- A ExecutableDeviceComponent may realize the LoadableObject interface
– optional Loadable UOF

Appendix F

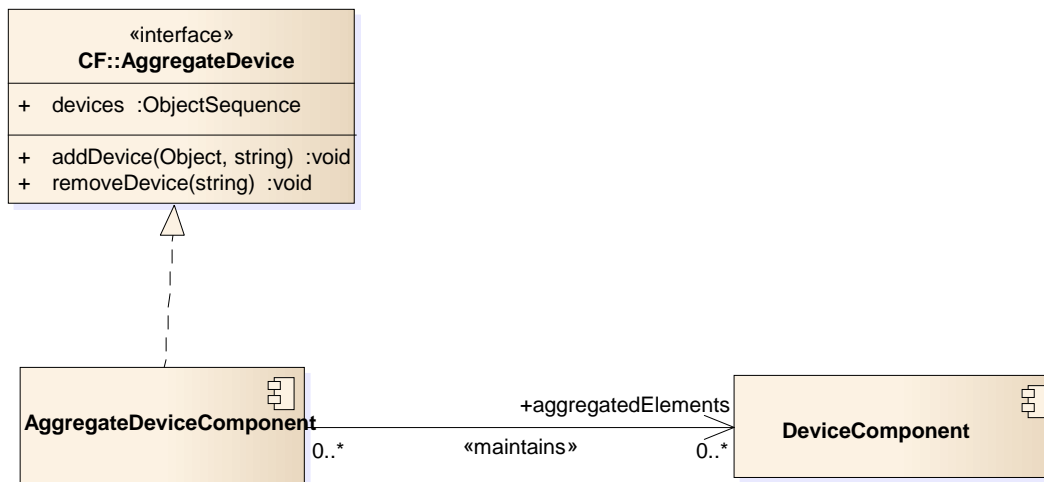
- Add Optional Loadable UOF for Executable Device Component

ExecutableDevice Component Figure



AggregateDeviceComponent

3.1.3.4.2.5 Change ComponentBaseDevice to DeviceComponent throughout section and update figure.



Platform Component Changes

Platform Component

PlatformComponentFactoryComponent

Service Component

CF_Service Component

Platform Component

Description

- A PlatformComponent is an abstract component utilized by the SCA Base-Device Components and Framework Services Components.

Update Figure 3-65

Added Association to FullComponentRegistry interface

- fullRegistrar: A PlatformComponent may be associated with a full registrar for unregistering itself.

Semantics

- SCA298 A ~~ComponentBaseDevice~~ PlatformComponent shall register with its launching DeviceManagerComponent via the *ComponentRegistry::registerComponent* operation by using the Component Registry execute parameter.
 - Tied to optional Component Registration UOF

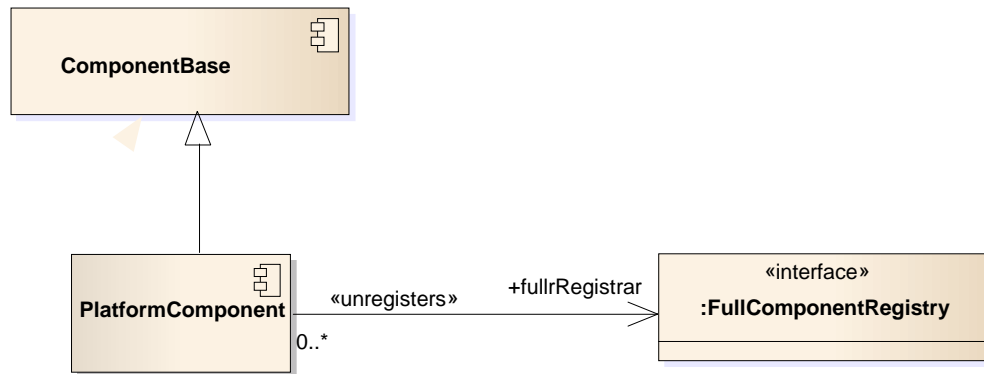
Add Constraints

- SCA_TBD A PlatformComponent shall fulfill the ComponentBase requirements
- SCA_TBD Each PlatformComponent shall be accompanied by a Software Package Descriptor (SPD) Domain Profile file per [section 3.1.3.6](#)
- A PlatformComponent may unregister itself via the *FullComponentRegistry* interface by using the Component Registry execute parameter.
 - Optional Component Un-Registration UOF

Appendix F add mapping to new optional requirements

- A PlatformComponent shall unregister itself via the *FullComponentRegistry* interface – Component Un-Registration UOF

Platform Component Figure



PlatformComponentFactoryComponent Changes

Semantics

- Remove “SCA412 A PlatformComponentFactoryComponent shall register with the launching DeviceManagerComponent via the ComponentRegistry::registerComponent operation.” since redundant with ComponentBase and PlatformComponent.

Constraints

- Add Req
 - SCA_TBD A PlatformComponentFactoryComponent instantiation shall fulfill the PlatformComponent requirements
- Modify Req
 - SCA416 The PlatformComponentFactoryComponent shall only launch ComponentBaseDevices or ServiceComponents

Service Component

Update Figure 3-67

Removed associations since moved to
ComponentBase.

3.1.3.5.2.6.3 Semantics

- Add a requirement. SCA_TBD A ServiceComponent shall set its registeringComponent type to be NON_CF_SERVICE_COMPONENT.

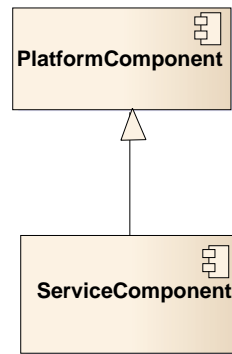
Remove Constraint

- SCA316 A ServiceComponent shall register with the launching DeviceManagerComponent via the *ComponentRegistry::registerComponent* operation

Add Constraint

- SCA_TBD A ServiceComponent shall fulfill the PlatformComponent requirements

Service Component Figure



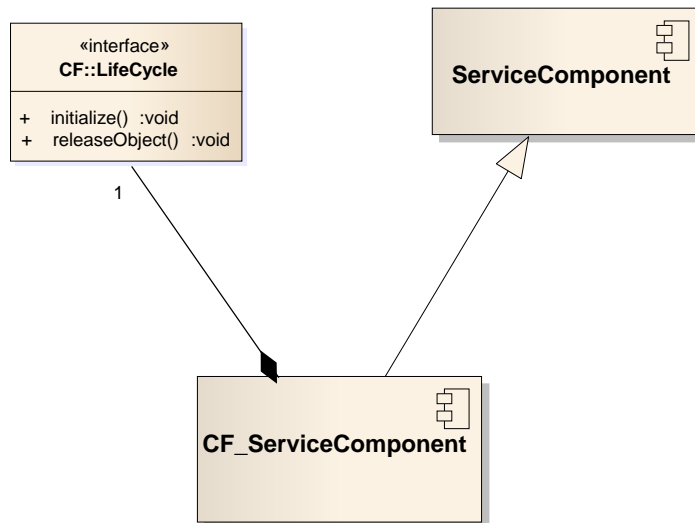
CF_ServiceComponent

Change Figure 3-68 to replace ResourceComponent with ServiceComponent, and realize interface to be LifeCycle.

3.1.3.5.2.7.4 Constraints

- Remove SCA529 A CF_ServiceComponent shall fulfill the ResourceComponent requirements.
- Add a requirement. SCA_TBD A CF_ServiceComponent shall set its registeringComponent type to be CF_SERVICE_COMPONENT.
- SCA_TBD A CF_ServiceComponent shall realize the *LifeCycle* interface.

CF_ServiceComponent



Appendix F Changes

F.6.2 ComponentBase Units of Functionality

- Update Figure
- LifeCycle UOF is now optional
- Added UOF - •Releasable – provides a release capability via the LifeCycle::releaseObject behavior; a component that is releasable

F.6.3 Application Related Component Units of Functionality

- Update Figure
- • ~~ResourceComponentApplicationComponent~~ is based upon ComponentBase UOFs
- AssemblyControllerComponent extends the ~~CF_ApplicationResourceComponent~~ with mandatory Controllable UOF
- • ~~CF_ApplicationResourceComponent~~ extends the Application~~ResourceComponent~~ with mandatory AEP UOF.

Appendix F Changes

F.6.4 ComponentBaseDevice Units of Functionality

- Update Figure
- Remove Device Releasable
- Replace text “ComponentBaseDevice” with “DeviceComponent” throughout text.
- • DeviceComponent~~BaseDevice~~ extends ComponentBase UOFs witha mandatory ~~DeviceReleasable-LifeCycle~~ UOF
- • ExecutableDeviceComponent extends the ~~Loadable~~DeviceComponent with mandatory Executable UOF

Figure 2: Component Base Units of Functionality Changes

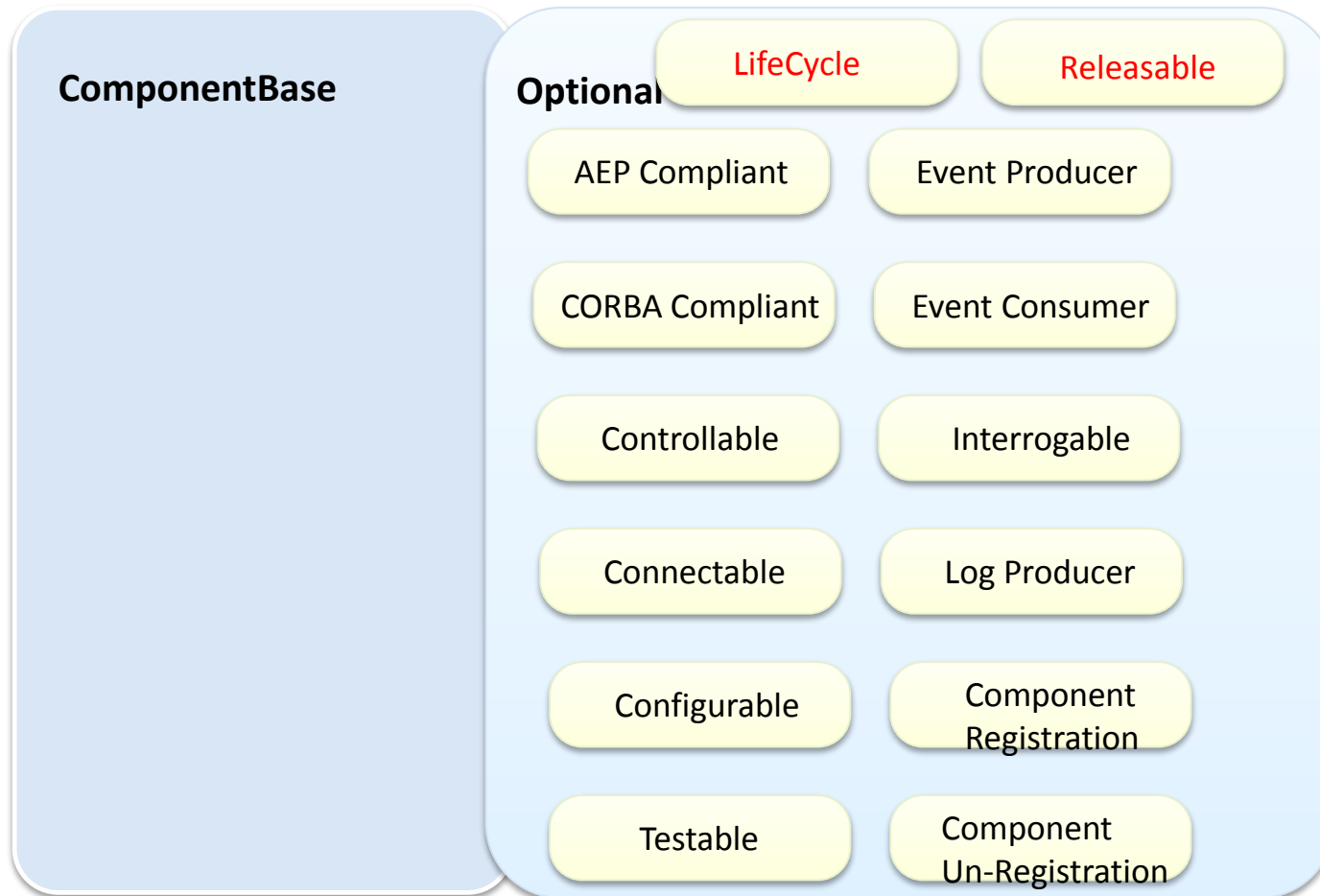


Figure 4: Device Components Units of Functionality Changes

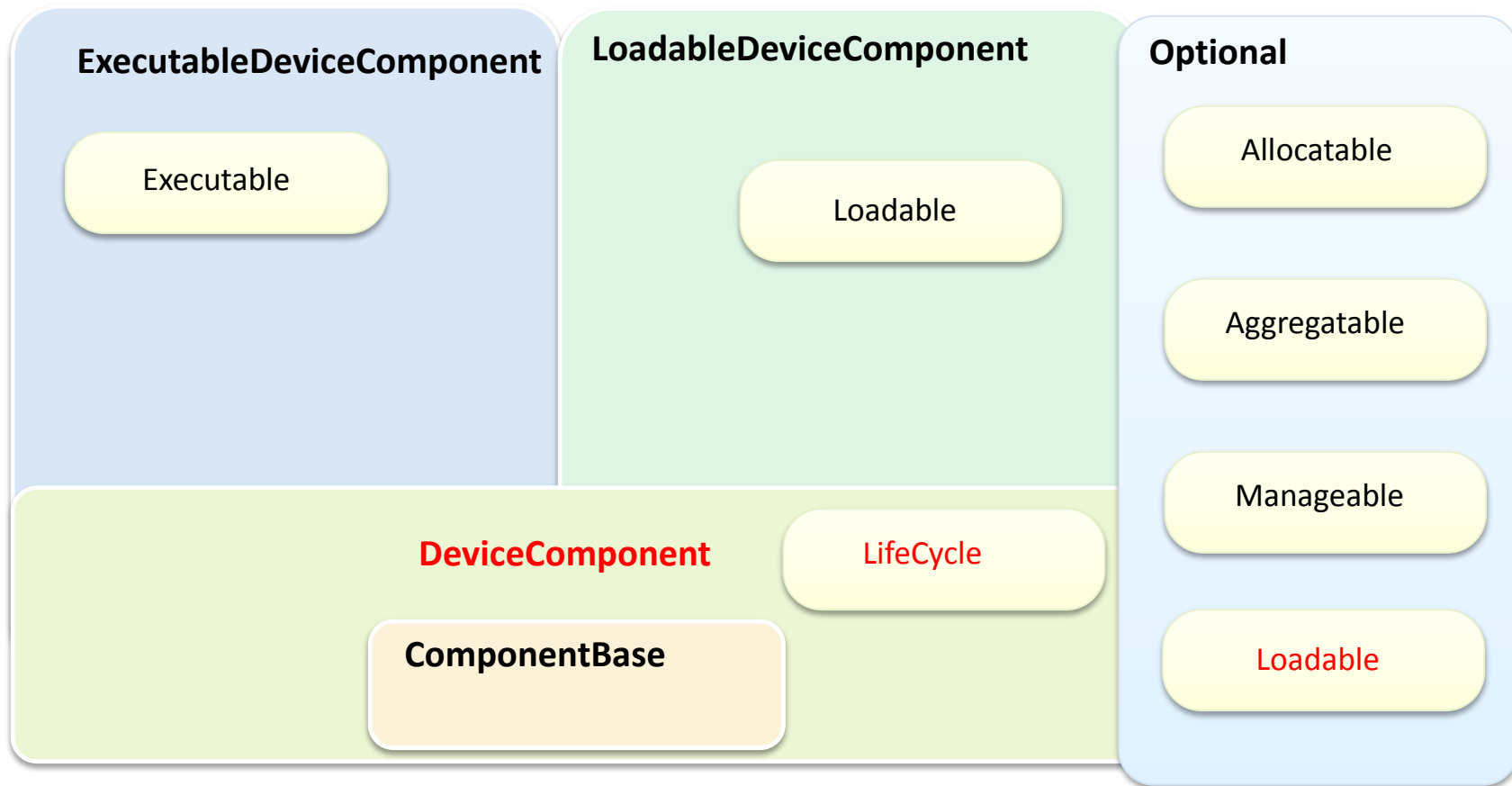
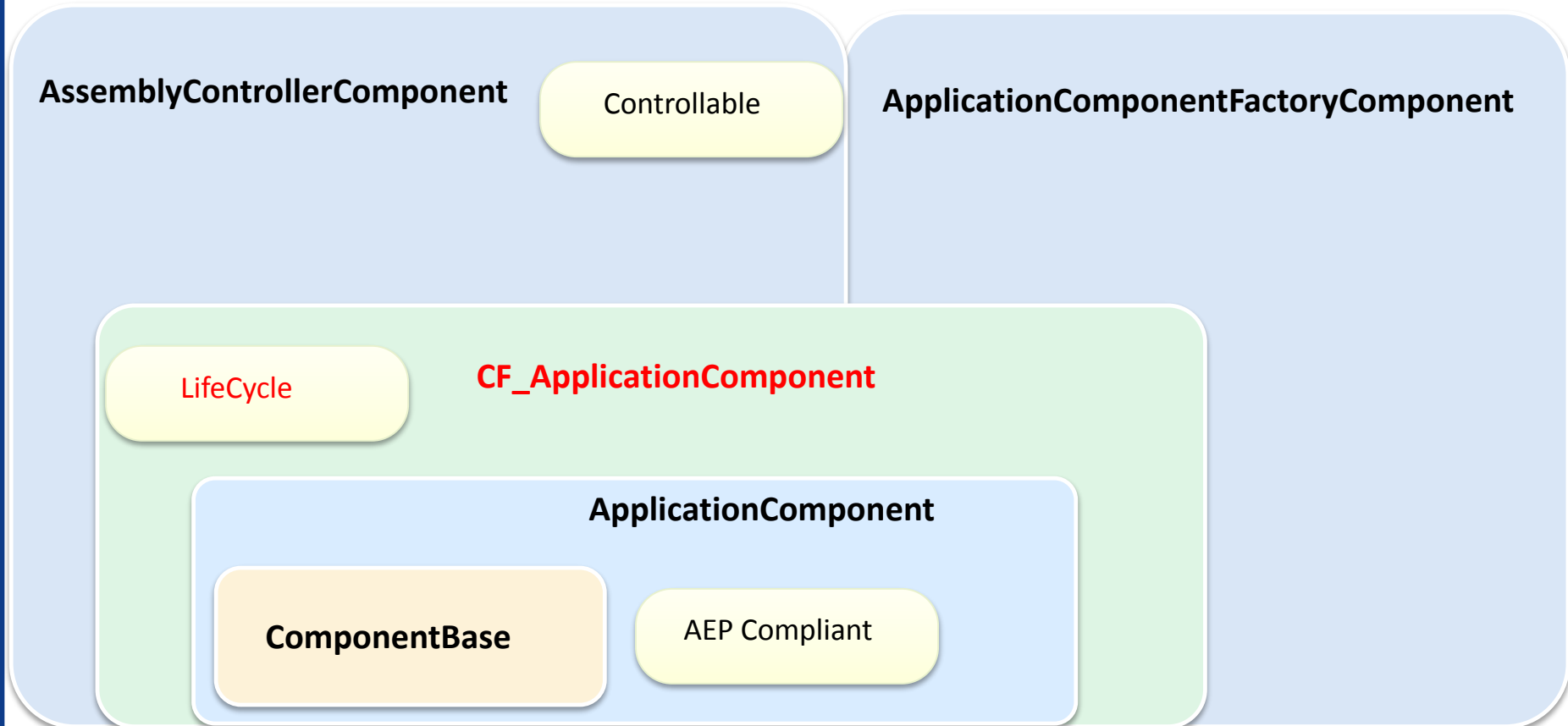


Figure 3: Application Components Units of Functionality Changes



Appendix C Changes

Remove Device, Resource Interfaces

Remove LoadableDevice

**Remove inheritance and rename ExecutableDevice
to ExecutableObject**