Feature areas and what folks are working on

This table is an attempt to capture the various requirements that I've gathered from browsing the SCA specs and discussions on the Tuscany dev list, and the work in-progress in the community.

We are hoping that it will help organize our requirements and discuss how we want to stage their implementation over our next milestones. It should also help new contributors understand who is working on what, and grab areas that need work.

The table is under construction. We are inviting everybody in our community to come help build that page with the requirements that they are aware of, and the features that they are interested in.

Feature area	Milestone	People working on it	Comments / description	Status
Assembly PM spec				
Complex properties		Venkat	Add support to SCDL parsers, builders and factory configuration. Integrate with data bindings.	This has been completed and checked into the sca-java-integration branch.
Pass by-value / by-reference		Venkat	Initial implementation almost in? Some cleanup necessary? Need to understand how to clone/copy data with different databindings, SDO, JAX-WS, and POJOs. Understand how to handle pointers across multiple parameters when copying.	Has now been improved for performance to skip copying when the source or target is a ServiceBinding or ReferenceBinding
Multiplicity			Support multiplicity constraints on references. Support multiple wires. Handle overrides. Investigate how to handle binding selection. Support for callbacks.	
Naming constraints in a composite		Luciano (Tuscany-914)	Validate that service/reference/component names are unique inside a composite. Check the presence of SCDL includes.	
Multi-valued properties		Venkat	Add support to SCDL parsers, builders and factory configuration. Integrate with data bindings.	This has been implemented and checked into the sca-java-integration branch. This impl. assumes that if a property is of simple type and has multiple values, then each value is defined in the scdl within an enclosing element who name is 'value'. e.g. <value>Some String Value< /value>.
WSDL 2.0			Add support for WSDL 2.0 accross the board, runtime, WSDL/Java tools and WS binding.	
Operation overloading (simple exact matching)		Rick	Change runtime service contract hashmap and invocation handling /dispatching. Adjust extensions to this. Match identical methods on ends of a wire.	
Operation overloading (complete support)			Match and wire compatible methods, factor in different databindings, WSDL wrapping/unwrapping, and match invocation and actual parameters to a compatible method.	
ComponentType side files			Partial support already in. Need to define overriding strategy when implementation and componentType metadata overlap.	
Support for SCA contribution		Raymond, Luciano	Generic handling of SCA contributions and base plugin mechanism for loading/scanning artifacts. Allow a contribution and artifacts in it to be addressed by URI. Do not assume a fixed default.scdl file.	The following is a summary of what's available for the contribution service: - Contribution Model - Processing jar and folder based contributions - Artifact loader WSDL, XSD, SCDL/Composite, Java/Class, sca-contribution.xml - Contribution repository - Contribution Service (add to SCA Domain using Assembly services) - Integration with SCAMiniRuntime, unit test cases, and integration tests cases - Very initial artifact addressing/resolving integrated into Assembly services - Support for loading sca-contribution.xml Design Draft Look for the following jira issues to see what's still not available in the contribution services : TUSCANY-1158, TUSCANY-1159, TUSCANY-1160, TUSCANY-1161, TUSCANY-1162, TUSCANY-1163
Filesystem based SCA contributions		Raymond, Luciano	Support free-form folder structure in a contribution. Support multiple SCDLs per contribution. Scan for SCDL and other artifacts under an SCA contribution.	Completed and available in the sca-java-integration branch.
JAR based SCA contributions		Raymond, Luciano	Support JAR contribution structure. Support multiple SCDLs per contribution. Scan for SCDL and other artifacts under an SCA contribution.	Completed and available in the sca-java-integration branch.
SCA includes			This does not seem complete, verify, in particular wiring across includes and support for nested includes.	
Composite resolution (no recursion)		Luciano	Register composites and find them by Qname within a contribution. Support references across contributions in domain level includes.	
Composite resolution (recursive composition)			Resolve composites with nested composites. Support full recursion. Current support needs to be fixed, as each reference to a composite gets it reloaded/redefined.	

XML property configuration in side files			Properties configured in an external XML file. Resolve the file location, load it. See if loading and xpath reference mechanism needs to be adjusted for this.	
Ref/Service/Property config override (no recursion)			Support overriding of services/references/properties. Investigate overrides combined with multiplicity.	
Ref/Service/Property config override (recursive)			Support overriding of services/references/properties in nested composites. Handle multiple overrides. Support additions vs overrides in the case of bindings. Investigate overrides combined with multiplicity.	
Wiring with service/reference bindings			Support binding URIs. Then configuration of binding URI with wires. Then propagation of binding info from a service to a reference wired to it.	Support for WS binding mainly completed determining the URI from all the combinations described in the assembly and WS binding specs.
Support for business exceptions		Rick, Raymond	Distinguish handling of runtime exceptions vs business exceptions. Add support to invocation/dispatching mechanism + POJO implementations, bindings and databinding.	We added the support for intra-composite and inter-composite over binding.axis2. Please see the Design Draft. There are a few itest cases you can play with. • Exceptions test • Cross biniding test • WebService binding test
Context info in error reports			Across the board, need to provide application level context data (composite, component, service, reference etc.) with errors, to allow an app developer to understand what's wrong in his application. Without that he's left having to understand all the underlying middleware.	
Minimum SCDL extensibility (non SCA namespaces)			Support additional bindings, implementation types, interface types, and policies from diff namespaces. Should already work for bindings and implementations. Need to check policies and interface types.	
Complete SCDL extensibility as defined by the XSD			Will need changes to a number of loaders and builders.	
Support for xsi:type in addition to global elements			Changes to most loaders or improvements to LoaderRegistry.	
Support for various ordering of SCDL elements		Simon	Loaders depend on a specific order of elements in the SCDL files. Supporting other possible sequences will require changes in the loaders logic.	
Autowire - local			Adjust to the spec, which now defines this at the assembly level, independent of the Java C&I.	
Autowire - domain level			Support autowire in a domain, including components services and references, policy intents and compatible bindings.	
Autowire extension point			Plugin mechanism to allow different autowire algorithms to be plugged in	
SCDL validation, semantic constraints (obvious cases)			Only support the obvious/main semantic constraints. Separate tool to perform semantic validation of an assembly. Also needs to be invoked as part of deployment and/or loading at runtime.	
SCDL validation, semantic constraints (complete)			Complete support for the spec. Separate tool to perform semantic validation of an assembly. Also needs to be invoked as part of deployment and/or loading at runtime.	
Implementation / component matching			Make sure that an implementation matches the component that uses it. For example enforce wiring of required references.	
Callback support		Ignacio	Combinations of sync and async. Determine which binding to use for a callback invocation.	
Configured implementations			Configuration of component implementations in .componentType files or inlined in implementations (spec issue 8) and in component declarations. Resolve/merge when overlap between the three.	
Promotion of Ref/Services/Properties (no recursion)			New promotion mechanism for services/references/properties. Service/reference/property definition on components. Handle promotion of already wired services and references.	

Promotion of Ref/Services/Properties (recursive)			New promotion mechanism for services/references/properties. Service/reference/property definition on components. Handle promotion of already wired services and references.	
Multiple bindings on services and refs			Basic support for multiple bindings is in. See how configuration overriding works with multiple bindings.	
Ability to use and alter SCDL model at deployment		Sebastien	Add missing relationships to the SCDL logical model, independent of runtime context. Allow deployment tools to alter the model and write it back to XML.	
Interchangeability of Java and WSDL		Ant, Rick	Build ServiceContract representation from Java or WSDL. Needs more work to understand combinations of SDO/JAXB/POJOs and wrapped/unwrapped. Add logic to wiring framework to match WSDL portTypes and Java interfaces.	Support for using interface.java or interface.wsdl interchangably completed. See TUSCANY-1111 for some limitations, mainly due to limitations of the databinding framework.
Document limitations and deviations from the spec			Review the spec and document Tuscany limitations and differences with the SCA assembly spec.	
Java C&I spec				
Derive service name from interface name			Clarify with specification, is the service name the fully qualified class name, the class name without the package name, or are we adding a service name attribute to the @Service annotation	
Injection of multivalued properties		Raymond	Map multi-valued properties to java.util.Collection or [].	Supported by sca-java-integration branch now.
Injection of references with multiplicity 0..n		Raymond	Map references with Multiplicity to java.util.Collection or [].	Supported by sca-java-integration branch now.
Required properties			Validation that required properties are configured	
Service references (minimum)			Minimum implementation of Service references supporting setCallback/getCallback. Local usage only with no serialization of the reference.	
Service references (complete support)			Complete support for the Service reference programming model, including the ability to pass Service References around, serialize them, then use them to perform invocations outside of the context where they were initially obtained.	
Conversational			Not sure what's missing. Check the spec against the runtime to assess what's remaining first.	
WSDL2Java / Java2WSDL / SDO - simple cases			Command line tooling. Support wrapped/non-wrapped patterns, multiple parts/message, nested complex types, faults/Exceptions. Generation of SCA annotations in interfaces.	
WSDL2Java / Java2WSDL / JAX-WS			Command line tooling. Wrap/integrate the JAX-WS tooling. Customize it to generate SCA annotations in interfaces? Investigate how the runtime could reuse the JAX-WS WSDL/Java mapping logic as well, since it'll need to know the mapping rules as well.	
@Autowire			Adjust to what the latest spec, maybe a little different from the current Tuscany support.	
Configured implementations			Add support for Java annotations allowing configuration of a POJO (services, references, bindings and properties) without SCDL.	
SCA as a wrapper over JAX-WS async			Allow cast of an SCA proxy to a JAX-WS API to support the JAX-WS async programming model.	
Ability to run with Java 2 security enabled			This is not just related to the Java C&I. Need to go over the runtime code and wrap sections of code that access resources, threads, classloaders etc in doPrivileged blocks to allow the runtime to work in an environment with Java 2 security enabled.	
Document limitations and deviations from the spec			Review the spec and document Tuscany limitations and differences with the SCA Java C&I spec.	
Conversational PM				

Core support		Ignacio, looking for volunteer to take over	Lifecycle in. Persistence of conversation state in. Needs to support tran/recovery. Also need some redesign to avoid having to send /receive a routing path and play nicely with bindings.	
Integration with bindings			Plugin mechanism to allow a binding to declare conv capability and support conversational. Maybe define a "conversational policy intent"	
WS binding		Rick	Map conversational to the correct MEPs. Investigate how to flow conversation id. Use WS-addressing, message-id, relates-to etc.	
JMS-binding			Need to understand how/if conversational means anything to the JMS binding	
EJB binding			Check the EJB binding spec to understand what needs to be done here	
JSON-RPC binding			Investigate if it makes sense to support conversational with this binding	
SCA default binding			Understand how a concrete binding will be selected depending on conversational requirements (may fold into some policy intent work)	
REST binding			Investigate if it makes sense to support conversational with this binding	
HTTP Session support			Persist session scope and conversational state with the HTTP session?	
Non-blocking PM				
Core support		Ignacio, looking for volunteer to take over	No support for async over sync yet (see spec issue) but everything else seems to be there. I am concerned with how we represent callbacks and how their info flows between remote components	Async one-way and with callback work. Integration tests in the sca-java-integration branch are successful. We'll need to revisit later how to represent callback endpoints when flowing over bindings, in particular the Web Service and JMS bindings.
Integration with bindings			Plugin mechanism to allow a binding to declare support for async and callback, MEPs and/or polling.	
WS binding (short term /sync MEP)		Ignacio, looking for volunteer to take over	Dispatch call to a thread and continue to use a sync MEP. Will not support unsolicited callbacks. Issue: WSDL representation of callback vs Axis2 client and sync MEP.	
WS binding (real async MEP)			Use an async MEP and WS-addressing and callbacks. Also investigate what needs to be done w.r.t security on the callbacks.	
JMS binding			Sync up with the latest JMS binding spec	
EJB binding			Do we need to support this? We support async invocation of POJOs, but people know that EJBs are synchronous.	
JSON-RPC binding			Used in AJAX apps which are based on async exchanges. So the JSON-RPC binding should support the non-blocking PM and callbacks.	
REST binding			Not sure if there is a requirement at all for real CRUD REST. There may be a requirement for activity based XML/HTTP services, similar to the WS binding requirement.	
SCA default binding			Understand how a concrete binding will be selected depending on non-blocking invocation requirements.	
SCA default binding				
Client side				
Server side				
Selection of concrete binding			Based on policy intents, capabilities of the installed runtime, and where the target service is deployed.	
JMS binding				
Adjust to latest SCDL spec			Support for the latest SCDL syntax. Support for JMS in a JMS managed environment.	
JMS data binding support			Support data bindings as defined in the JMS binding spec.	
Integrate with Tuscany data bindings			Integrate with Tuscany databinding framework. Support sending /receiving JAXB or SDO objects through JMS.	
WS binding				
Adjust to latest SCDL spec			A few changes to adjust to the latest WS binding spec.	
Configuration of WSDL2Java mapping			Related to the Java / WSDL story. Use the SDO based and JAX-WS Java / WSDL generators to produce WSDL? Apply the corresponding mapping in the runtime as well.	

Handling of faults			Support faults at least with SDO and JAX-WS. Support other data bindings available through the data binding framework.	
Support for attachements				
Support for RPC encoded				
Working without WSDL			Implement the JAX-WS defaults in the absence of a WSDL definition.	
Support for wrapped/non-wrapped			Add support for JAX-WS wrap/unwrap patterns. Apply the same principles with SDO and other data bindings.	
Support for SOAP headers			Investigate how/if SOAP headers should be exposed to a client and target service.	
EJB binding				Under IP-CLEARANCE process.
Client side				
Server side			Generate of EJB "wrapper" code and EJB DD?	
Geronimo integration			Generate Geronimo specific artifacts, and/or tighter integration with the Geronimo runtime, potentially avoiding any code generation.	
REST binding				
Resource interface and PM			Define a new component interface type for CRUD operations on a resource	
Client side			Mapping to HTTP verbs on the client side	
Server side			Dispatching of HTTP verbs to a component on the server side	
Activity style interactions			Not real REST, similar to the WS binding but with plain XML / HTTP	
Support for HTTP headers			Support for last-modified, etags and other HTTP headers	
JSON-RPC binding				
Support for references			Currently only supports services. Needs to add client side support as well.	
Integrate with latest toolkits			Integrate with latest JSON toolkits, initial integration was done some time ago, we need to refresh it.	
Complete support for JSON spec			Check what's missing in the current runtime	
Data binding support		Raymond		We have made quite a few improvements in this area in the sca-java-integration branch. • Added introspection of java interfaces to recognize known types for various databindings • Improved SDO and JAXB handling • Support for exception transformations accross databindings • Added JavaBean2StAX transformation • Improved StAX related transformations • Added more itest cases
SDO and JAXB			Support conversions between SDO and JAXB	
AXIOM support			Support conversion from/to AXIOM	
SCDL annotations			Define SCDL annotations to configure data bindings. Needs to be pluggable, maybe change at some point to use Intents	
Add more transformers and perf optimizations			The databinding framework approach will perform only if we have enough transformers to reduce the number of conversions going from one format to another.	Added the SDO --> AXIOM transformer using OMDataSource
Policy Intent framework				
SCDL and core support, model, loaders, builders				
Intent determination algorithm				
Wire matching				

Binding / container configuration				
Transaction policy				
SCDL model, loaders, builders				
Runtime interceptors				
Transaction service integration				
Integration with bindings				
WS-binding support for transactions				
JMS-binding support for transactions				
EJB binding support for transactions				
SCA default binding support for transactions				
Security policy				
SCDL model, loaders, builders				
Runtime interceptors				
Authentication / autorization service integration				
Integration with bindings				
WS-binding support for security				
JMS-binding support for security				
EJB binding support for security				
SCA default binding support for security				
JSON-RPC binding support for security				
REST binding support for security				
Reliability policy				
SCDL model, loaders, builders				
Runtime interceptors				
WS-binding support for reliability				
JMS-binding support for reliability				

SCA default binding support for reliability				
Modular build				
Extensions			Refactor extensions to improve our build system and allow the various parts of the project to be built independently	
Samples			Refactor samples to improve our build system and allow the various parts of the project to be built independently	
Integration tests			Refactor integration tests to improve our build system and allow the various parts of the project to be built independently	
Packaging / deployment				
SCA domain repository / metadata		Raymond, Sebastien	Define services to add/remove composites and maintain a logical SCA domain level composite. Needs to be pluggable. Support domain base URI and base URI for each binding scheme.	
Deployment of SCA contributions		Raymond, Luciano	Define services to add/remove SCA contributions to the SCA domain.	
Artifact addressing, resolving, loading		Raymond, Luciano	Plugin mechanism to handle addressing of artifacts within contributions, resolution within a contribution, conflicts between contributed artifacts and externally referenced artifacts (WSDL and XSD for example)	
Deployment extensions		Raymond, Luciano	Plugin mechanism to allow an extension to contribute to deployment as well.	
Tuscany classloading infrastructure			Support classloaders per SCA contribution. Isolate the application artifacts from the Tuscany runtime artifacts.	
OSGI classloader integration				
Administration				
JMX support			Basic services start/stop etc. Need to assess what's in the current code base.	
Diagnostic / event logging			Go through the kernel code and add event logging plug points.	
Monitoring / instrumentation			Go through the kernel code and add monitoring plug points. Add monitor plug points on composite, component start/stop, service invocation etc. and make sure that they are given enough context.	
Core start /stop/query admin services			Basic core services to start/stop/query running components.	
Scenarios				
Nested composition				
Domain level wiring				
Domain wiring across runtimes				
Mixing runtimes in a composite				
Web application integration				
Mediation / routing				
Web 2.0 application				
Adjust Bigbank to the latest spec				
WS-I SupplyChain application				
Host runtimes				
Tomcat integration			Our M2 integration story packages the whole runtime in each web app. We need another better / lighter integration.	

Geronimo integration			Just run Tuscany webapps on Geronimo	
Geronimo integration			A real integration with the Geronimo runtime	
OSGI integration				
Standalone server				
Spring component type				
Extend Spring to declare SCA services / refs			This is described in the SCA for Spring spec	
Use a Spring assembly in an SCA composition			Looks like a subset of the SCA for Spring spec is implemented?	
Javascript component type				
Support for JavaScript components				
Groovy component type				
Support for Groovy components				
JRuby component type				
Support for JRuby components				
BPEL component type				
Support for BPEL components				
Function test suite				
TBD				
Integration test suite				
Complex composition test cases				
Multiple runtimes / containers / bindings				
WS based test cases				
EJB based test cases				
JMS based test cases				
REST based test cases				
JSON-RPC based test cases				
Spring integration test cases				
Samples				
Sample binding				

Sample databinding				
Sample component impl. type				
Sample policy				
Simple assembly samples				
WS binding samples				
JMS binding samples				
EJB samples				
JSON-RPC samples				
Nested composites				
SCA default binding				
SCA include samples				
Property samples				
POJO samples				
Jruby samples				
Javascript samples				
Groovy samples				
REST samples				
Configuration override samples				
Transaction samples				
Reliability samples				
Security samples				
Standalone server samples				
Web app integration samples				
Spring integration samples				
Architecture documentation				
Kernel				
POJO support				
Plugging in a component impl type				
Plugging in a binding				
Plugging in a policy				
Plugging in a data binding				
Plugging in an interface type				

Plugging in a new host integration				
Building				
JMS binding				
WS-binding				
JSON-RPC binding				
RMI/EJB				
Deployment / packaging				
OSGI integration				
Spring integration				
User documentation		Shelita		
Build Tuscany				
Develop an SCA app				
Build an SCA app				
Package /Deploy				
Run				