Inspection Checklist - Software Architecture Document
From Software Architecture Document Guidelines, by Simon Brown, edited based on project scope.
Inspector: Matt Fowler

	Comments/Notes
Functional View	
Is it clear which features/functions/use cases are significant to the architecture?	Yes, very clear. I needed some clarification on the metadata lists which are clarified in the vision document
It is clear that these have these been used to shape and define the	Yes
Non-functional View	
Is there a clear understanding of the non-functional requirements that the architecture must satisfy?	Yes. You might want to clarify scale requirements such as load level, latency, and result size
Are the non-functional requirements quantifiable and testable?	Yes
Have common non-functional requirements been explicitly marked as out of scope if they are not needed (e.g. “user interface elements will only be	NA
Are any of the non-functional requirements unrealistic? (e.g. true 24x7 availability is typically very costly to implement).	No
Architectural Principles	
Are there any other principles (e.g. other non-functional requirements that have not been explicitly requested) that have helped influence the	No
Architectural Constraints	
Are the constraints well documented and comprehensive?	Yes
Is it clear how the constraints affect the architecture?	Yes
Process View	
Is it clear what the system does from a process perspective?	Very clear
Are the major flows of information through the system well understood and documented (e.g. using UML activity diagrams)?	Yes, very well communicated
Logical View	
Is a logical view of the architecture clearly portrayed?	Very clear
Does it show the major components and interfaces?	Yes
Are they described at a high level?	Yes … description contained in naming
Does the logical view show external systems and any other dependencies at a high level (low level detail about the dependencies isn’t required here)?	Yes

Interface View	
Are the key internal (e.g. databases, messaging systems, etc) and external interfaces (e.g. other systems) well specified at a high level?	Specified OK
What format are the messages (e.g. plain text or XML defined by a	WSDL
Who has ownership of the interfaces?	Not clear
Technology Selection	
Is it clear why the selected technologies were chosen?	Yes, in the notes
If there were options, why were they not chosen?	No other options addressed
Do they all fit in with the constraints outlined previously?	Yes
Are all software and hardware tiers covered?	Yes, implied.
Design View	
Is it well understood how the key use cases will be implemented?	Yes
How are the chosen technologies used and combined?	Package structure shows excellent view and relationships of technologies. Very standard
Are there common patterns across the architecture?	Yes
If yes, are these well understood and documented?	Yes
Are the diagrams (e.g. UML class and sequence) up to date and do they	Yes
Are any common wheels being reinvented? If so, why aren't vendor/open source products being used?	No. Very good use of existing technologies, like Spring and Hibernate
Is there enough information here to provide the rest of the development team with an overview/the intent of how the designs work?	Yes
Infrastructure View	
Is there a clear physical architecture?	Yes
What hardware does this include across all tiers?	Hardware not clear, but implied in database and application tiers.
Does it cater for redundancy, failover and disaster recovery if applicable?	Yes, definitely. Clustered environment fronted by load balancer meets all three
Security View	
Is there a clear understanding of how security is handled within the architecture and how any security requirements have been satisfied? This	
• Authentication.	Not applicable to the scope of this project

• Authorisation.	Not applicable to the scope of this project
• Confidentiality of data between components (e.g. during user login, during requests between components, using technologies such as web services or messaging, across public networks).	Yes, very clear. Metadata that is shared does not contain sensitive data.
• Different types of users and their roles.	
• Network separation using firewalls and DMZs (red, amber, green model).	Yes
• Restricted access to resources.	Yes
• Permissioning of data of a per user/role/etc basis and the ability to modify those permissions.	NA
Monitoring, Management and Administration View	
Is it clear how the architecture provides the ability for operation/support teams to monitor and manage the system?	Yes
How is this achieved across all tiers of the architecture (e.g. from client tier	Primarily, logging
Data View	Excellent view that corresponds directly to conceptual view.
Is there a high level understanding of how much storage will be required to	No. Dependent on usage of the system
What are the archiving strategies?	Not specified
Are there any regulatory requirements for the long term archival of	Not specified
Likewise for log files and audit trails?	No
Justification of Non-Functional Requirements	
For each of the non-functional requirements, is it explicit how the	Yes, very clear
In the case of performance and scalability targets, are the test cases and results referenced?	Yes, again very clear
Are there any single points of failure?	No, all points are handled that I can see
What happens if a component fails?	Dependency on JGroups is clear.
Can you recover from a system failure?	Yes
Who is responsible for system recovery and failover?	Not specified