Infrastructure - Task #1035

Develop reference architecture documents

2010-11-04 19:48 - Bruce Wilson

Status:	In Progress		**Start date:**	2010-11-04
Priority:	Normal		**Due date:**	2010-12-31
Assignee:	Bruce Wilson		**% Done:**	30%
Category:	Documentation		**Estimated time:**	0.00 hour
Target version:				
Milestone:			**Story Points:**	
Product Version:	*			

Description

Per EAB, need reference architecture documentation, in a format that will satisfy Martha Maiden, in particular. A focus for this is helping to understand commonalities and potential differences in member nodes.

History

#1 - 2010-12-29 17:45 - Bruce Wilson

- Status changed from New to In Progress

Working on a higher level piece of documentation and some lower level docs.

#2 - 2011-03-03 19:26 - Bruce Wilson

Work with Amber on this. Needs a CE perspective on the CI. Start from the 15 pager to NSF. Look at this from a white paper perspective. 20 pages, not the 500 pages of the total architecture documentation.

A reference architecture is a high-level system design free of implementation details and consists of the following elements:

- a high-level description of the system components

- definitions of relationships between components

- definitions of relationships between system components and elements external to the system

- Identification of performance drivers and capacity requirements

The key issues addressed in this particular reference architecture are (a) definition of the overall structure and relationships between key processes (functional elements) of the system, indicating the flow of data and control between processes and (b) identification of key performance drivers (from the SSCS FRD), including computer power, network bandwidth, and data storage to ensure that sufficient system capacity is in place to handle the expected loads.

In addition, this architecture provides high-level definitions of key data sources, data stores produced, and interfaces between the system components. These interfaces are specified as functional Application Program Interface (API) definitions. This architecture also shows how the TT&C components relate to mission-unique components and data sources required to make a complete mission data processing system.

Finally, this architecture provides a notional distribution of SSCS functionality onto workstation platforms and a notional network topology. These were derived using performance benchmarks and analyses to determine the platform size needed for the driving requirements.