

Science Support View



Nominal Workflow



DFD Runtime Overview

dfd Runtime overview , Historical System states Where, What Maintenance task: Archive States States 1 States Calamities/Event Do Maintenance Status updates task Actual System Broken Hardware, SuperNova, Data Products states Task related events Handle Calamity/Even `Availability Status update Signal and Metadata Observation Task: System Events Task related events Status update System failures Events Resources and settings Handle system Moni tor Failures System State events Alarms Rules Run-state Do Alarm/Aler Software Observation task Notification Rule: Occupation Task Failures Taskinformation Guard Progress Continuity Hardware Progress Task progress <<Abort Task>> Data Quality Products Active tasks Finished tasks Created dataproducts and size Validate Quality Measure: Runnable Tasks Results Status update Realized Quality Proposal/Semeste DataProduct progress Information

DFD Planning Overview

dfd Planning overview 🦯





Reference architecture Motivation



- Dot on the horizon
 - Focus & guide for design of new and adapted functionality
 - Idem for future refactoring of existing components
- Best practices
 - Many roads leading to Rome
 - Some better than others, not always clear
 - Build on proven architectural design principles
 - Re-use knowledge, tools, etc.
- When detailing design for an aspect/some functionality
 - Map to reference/target architecture
 - Validate that requirements (functionality, performance) can be met
 - If necessary: motivated deviation

Reference architecture Requirements



- Data and event driven model
- Modular independent blocks
- Stateless process interfaces
 - Allow for diverse user interaction patterns
 - Keep status of components (in database)
 - Database access through data access component
- Loosely coupled interfaces
 - Properly documented API (validate) and version control
 - Unit testable
- Parallel handling of multiple requests
 - Scalability: Allowing multiple instances of the server/service

Reference architecture

Three tier architecture with message bus



Target LOFAR architecture



Message bus supports multiple communication channels

- Information exchange
- Notifications and alerts
- Logs
- ...
- Public interfaces via web-server
 - Many reusable tools & components available
 - No firewall issues etc.
- Strive for consolidation of databases and data access services
 - Prevent concurrency issues & inconsistencies
 - Reduce technologies that need to be supported
- Long term perspective, full implementation probably not in scope of 3R
- Feedback welcome, to be discussed in targeted follow-up meetings

Current LOFAR Architecture



Target LOFAR Architecture



Mapping DFD and Target Architecture



Next steps



- Discuss and finalize high level Target LOFAR architecture
- Map RRR objectives on Current and Target architecture
 - Identify migration steps
 - Reduce risk by making steps manageable (i.e. not too big)
 - Aim for best result with least effort
 - Do not phase out existing components until full replacement operational
 - Get closer to Target architecture
 - Engage stakeholders for detailed design
- Continue work to complete architectural documentation in parallel
 - User requirements, Performance indicators, Information architecture (DFD), Application architecture (as presented)
 - TBD Technical architecture