

Mero Architecture Documentation

Architecture Documentation - views and their contents

[Deployment of the documentation](#)

[Component Documents](#)

[Baselining](#)

[View Packets](#)

[General Structure of View Packets](#)

[Types of included view packets](#)

[The view packets in the C2 context.](#)

[Contents & structure of the view packets](#)

[Module decomposition](#)

[Layering module view](#)

[The uses module view](#)

[The generalization module view](#)

[Pipe and filter & Multi tier view](#)

[Shared data & publish subscribe views](#)

[Client server & communicating processes views](#)

[Deployment allocation view](#)

[Implementation allocation view](#)

[Work assignment allocation view](#)

[View packets for CSIM](#)

Introduction to C2

Beyond Views

[Stakeholders and their concerns](#)

[Development](#)

[Mappings between views](#)

[Considerations and background](#)

[Glossary](#)

C2 Core Architectural Views

[C2 Core - Module Generalization View](#)

[C2 Core - C&C Shared-data View](#)

[C2 Core - C&C Client-server and Communicating Processes View](#)

C4 Architectural Views

[C4-T0 Module Decomposition View](#)

[C4-T0 Allocation Deployment View](#)

CSIM Architectural Views

Mapping architecture to requirements

[Summary requirements for C2, C4 & CSIM](#)

[Detailed QAS for C2 Core](#)

[Detailed QAS for T0](#)

[Detailed QAS for CSIM](#)

Abstract

This document describes the structure of the C2 Architecture Documentation. This document is a prerequisite for contributing to such documentation and serves as a guide to the view packets that make up the core of the architecture documentation. It also contains a summary of the architectural and business goals of the C2 system, and a conceptual

overview of the system.

Organization of the C2 Architecture Documentation

This document has several parts, each part is subdivided into chapters.

1. The first part describes the deployment, structure and organization of the AD. We describe what view packets we employ, what is in them and how common design concepts in storage architecture map to the views we have chosen.
2. The second part is an informal description of the C2 business goals and architecture.
3. The third part is "Beyond Views". This contains a discussion of stakeholders and their concerns, a mapping between the views, a chapter with considerations and background information and a glossary.
4. Part 4,5,6 of this document contains sequences of view packets, as specified in the first part. At the head of these sections are directories for easy reference to where certain issues are discussed. Such sequences of view packets are found for each of the 3 major subsystems.
5. Part 7 contains a mapping to requirements. It contains a (long) table of summary requirements and in each has a reference to architecture that covers it. It also contains quality attribute scenarios and indicates where these are covered in the architecture.

The Architecture Documentation (AD) will be deployed as a Google site that includes google documents and google spreadsheets each containing view packets. The table of content of the AD is given above, and it contains sequences of each of the 12 view types. The views and component documents are described in this section.

Architecture Documentation - views and their contents

Deployment of the documentation

Component Documents

Baselining

View Packets

General Structure of View Packets

The documentation of C2 consists of view packets substantially similar to the example given in DSA, Chapter 10.2 and Appendix A (Chapter 2). Each view packet has the following sections:

1. Primary presentation
2. Element catalog
 1. Elements and properties of the elements

2. Relations and properties of the relations
3. Element Interfaces
4. Element behavior
3. Context diagram
4. Variability
5. Architecture background
 1. Rationale
 2. Analysis Results
 3. Assumptions
6. Other information
7. Related view packets

All diagrams in view packets shall be UML2 - note that DSA was written before UML2 was specified. UML2 does contain diagram types for all views.

Types of included view packets

The following view packets will be prepared for the AD:

1. Module decomposition view
2. Layering module view
3. Module uses view, including specification of interfaces
4. Module generalization view
5. Pipe and filter component and connector view
6. Shared data component and connector view
7. Publish subscribe
8. Client server
9. Communicating processes component and connector view
10. Deployment allocation view
11. Implementation allocation view
12. Work assignment allocation view

The contents of each section for each of these view types can be reviewed more extensively in DSA Chapters 1-5, and in the example in DSA Appendix A. A chapter in this introduction contains references to these sections or further specifications for notations and content.

The view packets in the C2 context.

In this subsection we will briefly survey the contents of each type of the view packets, in relation to C2. The following table maps well known names of elements of a distributed file system architecture to the view types where we will discuss these:

Architectural Element	View(s) containing its architecture
Client server protocols	Client server
Monitoring interface	Publish subscribe views
Programmatic interfaces for modules backend, request processing, management	Uses module view

Major subsystems	Layering view
Components of major subsystems	Module decomposition view
Product line, module types	Generalization view
Allocation algorithms	Shared data & publish subscribe
Disk formats	Shared data & publish subscribe
File Operation Packets	Shared data & publish subscribe
Analysis & Diagnostics Database	Shared data & publish subscribe
Filesystem Operation Log	Shared data & publish subscribe
Distributed transactions	Client server and communicating processes
Recovery	Client server and communicating processes
Layouts	Disk formats
Concurrency control	Client server and communicating processes
Monitoring data	Shared data & publish subscribe
Sequence diagrams & use cases	Client server and communicating processes

Contents & structure of the view packets

Module decomposition

The C2 system decomposes into 3 major subsystems: the core, C4 and the CSIM. The module decomposition further decomposes these subsystems. Examples of core modules are the daemons NFSd, CIFSd and C2d; the request scheduler NRS, the request handler RH etc.

Contents

Construct a primary view using UML package diagrams as on page 43, 44 of the DSA course slides.

See DSA Table 2.1 for a description of the elements, relations and properties. Use figure 2.1a

Module decomposition views will be included for the major subsystems, but also for complex modules, such as the request handler RH and for the client C2C

Layering module view

Layers consists of sets of related modules, e.g. backend modules, middleware modules and frontend modules. The layering relationship describes "allowed to use". This view further restricts through what kind of interfaces modules interact, and describes modifiability and portability quality attributes.

Contents

Primary view, use a UML 2.0 package diagram similar to slide 104 on the right. Read the comments to figure 2.11 in DSA, page 89.

See DSA table 2.4.

The uses module view

Most uses in C2 will be through interfaces; but other use does occur through sharing data. This section presents a list of the modules that interact and names and describes each interface the modules offer. These view packets are lengthy.

Contents

The primary view should use a component view with assembly connectors, as on page 61 of UML 2.0 In a Nutshell.

A section is included to specify the interfaces offered for use, when such use is through methods. See DSA section 7.3 for what needs to be written into an interface specification, and note that not every call of the interface has to be fully specified at the architectural level. These interfaces exclude the interfaces that consume events from FOL, ADDB and Monitoring data. Those interfaces are covered in publish subscribe views, the data structures underlying FOL, ADDB are in the shared data section.

Related views: This view parallels that of 3.1 in terms of organization into packets, by focusing on the elements offering the interfaces.

The generalization module view

This view indicates which modules are backends, middleware etc. It names the types of generalized modules, the generalized interfaces offered by these "classes" of modules. Examples are the metadata backend module and its interface.

Contents

Primary presentation will be a simple class diagram showing generalization as on the DSA course slides, page 49 (slide 97), or UML in Nutshell figure 2-28, page 28. However, when interfaces are implemented in multiple ways, use the notation from slide 99.

Pipe and filter & Multi tier view

We combine the two view types. The pipe and filter view shows how data flows through the modules the network or data stores, while it undergoes sorting, filtering and

transformations. The multi tier view shows very similar picture.

Contents

The primary presentation is a diagram as on DSA slide on page 138. Include a network pipe and a pipe for reading publish subscribe data.

See table 4.1 for definitions of the elements and relations.

Shared data & publish subscribe views

As suggested on slide 130 of the DSA course, we will combine these views.

This view discussed how multiple modules share data. The packets concern (1) the request handler, the pressure handler and subscribing data management applications sharing FOL disk data and (2) and multiple subsystems sharing resource data, (3) packets for ADDB subscriptions (4) exchange of monitoring data (5) the disk formats for data and metadata stores. These packets contain relatively detailed considerations for disk formats and will be long.

Contents

Note that there is no UML view for shared data or publish subscribe primary presentations. We will chose a diagram as on slide 122, with a clear legend.

Tables 4.2 and 4.3 indicate the elements that need to be documented in the element catalog. Under the element interface and behavior section, document the interfaces available for writing/publishing and reading/subscribing and allocation, as well as the architectural level facts about the record formats.

Client server & communicating processes views

These packets contain, mostly by reference, the protocols for NFSd, CIFSd and FUSE, and for modifications C2 optionally requires for these. A very central packet is a view packet that discusses the C2 request handler. Note that client and servers sometimes exhibit bi-directional (such as when including lock callbacks) it is better to document them under concurrent uses. Also note that servers like NFSd and CIFSd are not independent when NFS leases conflict with CIFSd oplocks or share modes.

These views are very central in the C2 architecture as much of the behavior and interaction with clients is encoded in these views. Each view will contain UML 2.0 state charts (with trigger guard effect indications as in Fig 8.12), sequence diagrams (note the fine points in diagrams 10.5, 10.12, 10.13, 10.14) and communication diagrams (figure 10.38). Activity diagrams may also be used, if they are clearer (see Fig 9.25). Be aware of the differences

between trace oriented documentation and non-trace oriented documentation (slide 163 and what follows).

We will have separate view packets for:

1. Overall view of the communications - this is very similar to the "conceptual architecture". Show running instances of modules and name the uses of communicating processes with a protocol name where possible.
2. C2 Request handler communications - with an emphasis on behavior to describe the internal behavior such as (1) pressure generation, (2) distributed lock management, (3) local concurrency control within the modules internal to C2.
3. C2 client communications - with an emphasis on the behavior for (1) clustered metadata, (2) server network striping (3) recovery
4. NFSd - emphasis on leases, recalls and recovery
5. CIFSd - emphasis on oplocks, share modes and recovery
6. FUSEd - emphasis on the state changes in the cache and interactions with the kernel.

Contents

The primary presentation displays the communicating processes in a UML 2.0 communication diagram (figure 10.38, UML in Nutshell), naming protocols where possible.

The element catalog walks through specific cases of type of interaction. Table 4.6 in DSA specifies the element catalog.

The interface specification supplements these with sequence and state diagrams documenting the behavior.

Deployment allocation view

This describes on which servers running instances of the software are to be deployed. It allows the performance quality attributes to be tracked.

Contents

See: DSA p457

In the primary description include a UML2.0 deployment diagram (see slide 147 of the DSA course materials) covering a general style of deployment, including all computer systems required by the deployment

Supplement this primary view with a table of target deployments: e.g. "Low End HS Servers", "High End HS Servers", "OEM XYZ servers with routers"

Include the following table with CPU, Memory, Storage, Network speed and capacity descriptions. One table for each target deployment

System	CPU type/speed	RAM capacity/speed	Network type, speed, latency	Storage capacity, throughput, buildup (mix of SSD, SATA, SAS)	Summary of configuration of the system

For each target deployment, state final target performance vector

Target Deployment	IO throughput / DS	MDOPS throughput / MDS	Capacity / DS	Capacity / MDS	SPECFS performance	Small File Perf

Implementation allocation view

This section describes programming language choice, source code organization and the owner of the module.

Contents

Primary Stakeholders: Implementers, Software Manager, Testers

Primary view - a UML 2.0 implementation diagram as on slide 152 of the DSA course materials.

The element catalog consists of the following table.

Unit #	Subsystem	Module	GIT directory	Language	File names	Owner	Comments

Work assignment allocation view

This describes which team works on the module, or how it is acquired from another provider. It includes size estimates.

Contents

Location: Chapter TBD

Primary stakeholder: Software Manager, Implementers

See: DSA p464

The primary view is a table with for each Implementation Unit (sometimes also called an

element) that is part of the architecture a row in the following table.

Unit #	Subsystem (Core, CSIM, C4)	Component Name	Development depends on units #...	Runtime: depends on units #...	Parallel Development Stream #	Development Size (LOC) & Effort (HRs) with variances, risk (L,M,H)	Estimate for Prototype	Assigned to org unit

Notes:

- Each parallel development stream number identifies a collection of units that can be developed simultaneously

In the element catalog, we have a row in the following table for each organizational unit:

Organizational Unit	Contractor or Internal Team	Point of contact (skype/phone/email)	Skill set (narrative)

Refer to Module Decomposition diagrams

Describe particular concerns, e.g. for software to be acquired

View packets for CSIM

The CSIM software is significantly simpler than the full C2 file servers and accordingly we will have fewer view packets. There is overlap with C2 view packets for ADP and ADDB shared data.

We will have the following view packets:

1. Module decomposition
2. Uses module view - this should document the restricted versions of the ADP interfaces required to run the simulator. It should also describe the interface to the core DES scheduler module.
3. Generalization module view - document the required abstract interfaces for modules to be added to the simulator.
4. Shared data / publish subscribe views - document how output is placed into ADDB efficiently for later processing
5. Communicating processes view - discuss the interaction with the scheduler
6. Deployment view - particular emphasis on partial parallelization of the simulator
7. Development allocation view - discussion of source, as above

Introduction to C2

This will be a reference to an updated version of the white paper.

You can find the latest version of white paper at [Dropbox/marketing/white-papers/2009-11/colibri-deepdive/2009-11-colibri-deepdive.pdf](https://dropbox.com/marketing/white-papers/2009-11/colibri-deepdive/2009-11-colibri-deepdive.pdf)

Beyond Views

Stakeholders and their concerns

Development

1. **Architects & Product Managers** Need to make sure this architecture is ready for implementation and conforms to the requirements
2. **Software Managers** Are concerned with the schedule and estimates for implementation of the architecture. Require dependencies and estimates for planning.
3. **Designers and Implementers** Need to know what to build, what they need to accomplish to implement the architecture, and when the implementation can enter test.
4. **Quality Assurance** Need to have sufficient information to perform QA and know when such is complete. One aspect of QA is checking that the implementation conforms to the architecture
5. **Testers** Does the AD supply sufficient information to perform architecture based test and when such test is complete
6. **Integrators** Need to know how elements of the architecture integrate with other software elements and with hardware
7. **Fielders** Are concerned with the deployment of the software at customer sites
8. **Customer & Program Manager** Need to know if and how specific customer issues are addressed by the architecture
9. **Benchmarkers** Needs to know the performance requirements on the units and methods to assess if they are met

TBD: stakeholders and different kind of views

Mappings between views

Considerations and background

[Glossary](#)

C2 Core Architectural Views

C2 Core - Module Generalization View

C2 Core - C&C Shared-data View

C2 Core - C&C Client-server and Communicating Processes View

C4 Architectural Views

C4-T0 Module Decomposition View

C4-T0 Allocation Deployment View

CSIM Architectural Views

Mapping architecture to requirements

Summary requirements for C2, C4 & CSIM

Detailed QAS for C2 Core

Detailed QAS for T0

Detailed QAS for CSIM