

Architecture

Overview of the systems structural elements and relationships based on a concrete architecture. Note that the documentation here describes a specific architecture chose based on the general [Protocol](#).

System overview

General overview of the system architecture and functionalities

ActiveMQ Message Bus

In the Danish national bit repository we use ActiveMQ brokers as the message bus

- [Message bus](#) — Description of the message bus we use in the Bitmagasin system
 - [Choice of message bus](#) — This is the rationale behind our choice of ActiveMQ as messaging software
 - [Message bus requirements](#) — The requirements our message bus solution needs to satisfy
- [Division of message content between xml and message headers](#) — Certain information must be accessible to the message bus in order to route the messages and for the clients and pillars to distinguish messages. On the other hand, for confidentiality and integrity reasons it is desirable that most information is exchanged within the signed and possibly encrypted xml in the message body.
- [Queues and topics](#) — Messages are sent over the message bus. For optimal performance, messages may be sent over dedicated queues and topics. This page describes a suggested use of queues and topics.
 - [Destination design decisions](#)
 - [Unresolved queue issues](#)
- [Coordination Layer](#) — The coordination layer is a logic container for all communication related concerns in the Bitrepository system.
 - [Policy for use - Coordination layer](#)
 - [Design Decisions - Coordination layer](#)
 - [Design Discussions - Coordination Layer](#)

File exchange

Security

Description of the the different security related aspects of the protocol

- [Operation permissions](#) — Describes how messages requesting operation on a collection is are checked against the permission model.
- [Encryption](#) — To prevent anybody from eavesdropping on the bit repository data and messages exchanged, all communication is encrypted. Two levels of PKI are used.
- [Authorization](#) — Authorization management is necessary because all clients/systems operating within a given SLA may not be allowed to have equal access to data.
- [Confidentiality](#) — Information about the Bitrepository confidentiality considerations

Stability

Defines the stability requirements for the participants of the system

Bit Repository instance

A instance defines a closed context in a BitRepository system for maintaining one or more collection of files.

- [Destination syntax](#) — Definition of the syntax for string representation of an ActiveMQ destination
- [RepositorySettings](#) — Describes the format of the RepositorySettings configuration.

Input to organisation

Design decisions - Architecture

Unresolved Issues - Architecture

Pillar

A pillar consists of a specific media and infrastructure giving it specific characteristics which are used when an agreement of service levels (SLA). A pillar is defined as a representation of a copy of data that can be seen and analyzed as an individual unit at the abstract level. Related to a given Bit Repository instance the pillar can either represent a Full copy pillar or a Checksum pillar. See also Bitmagasin pillars for more details of the specific pillars.

- [Pillar SLA](#) — Describes the pillar characteristics which needs to be defined, besides the ones defined in the RepositorySettings.