

Application design

Description of the design of the reference Java system developed as part of the project.

Reference architecture

The reference implementation of the system is shown on the right. The setup shown here includes:

- Specific pillars: The [Pillar](#) currently planned for the bitrepository.dk system.
- Infrastructure: The central infrastructure used in the [Coordination Layer](#).
- Reference service: The [Services](#) running in the system for monitoring and aggregation information. In case of the integrity service write operation are also available in case of corrupt files needing to be restored. The interface to the services will properly be message bus based, similar to the general protocol. Note that reference services are build on the same reference libs show in the endpoint client, but are now show.
- End-user client: A typical client used by the end-user. The reference libs can be used here for accessing the different functionalities in the system.

Modules

Description of the application breakdown into separate modules

- [MessageXML module](#) — XSD schemas defining the data structures of the Bitmagasin messages
- [Protocol module](#) — Contains the general protocol functionality, like message bus connectors, message processing and data transmission code
- [Access client module](#) — Contains functionality for reading from the BitMagasin. This includes Get File, Get Checksums and Get File IDs
- [Modifying client module](#) — All functionality for writing to the Bitrepository is located here. This includes Put File, Delete File and Replace File.
- [Processing client module](#) — Contains the client side functionality for Processing Single File and Mass Processing
- [Alarm client module](#) — Contains the Alarm client functionality
- [Integrity client module](#) — Contains the integrity client code
- [Audit client module](#) — Contains the client side functionality for accessing audit logs.
- [Monitoring module](#) — Contains the code for the central monitoring service, which is part of the Coordination Layer.
- [Integration module](#) — The tests for examining the integrated system (or part of it).
- [Common module](#) — Contains code shared by all modules. Examples are utils and configuration loading.