

openIDL V1 - Architecture - Functional and Non-Functional Requirements



UNDER CONSTRUCTION

This page is under construction.

- [Data Quality / Auditability](#)
- [Security](#)
- [Data Privacy](#)
- [Performance](#)
- [Scalability](#)
- [Availability](#)
 - [Inside Kubernetes](#)
 - [Outside Kubernetes](#)
- [Customizable](#)
- [Deployable](#)
- [Observable](#)
- [Usability](#)
- [Maintainability](#)
- [Monitoring](#)
- [Logging](#)
- [Support](#)

This page discusses the non-functional requirements and how they are met in the openIDL architecture.

Data Quality / Auditability

The data used as the source for extractions must be high quality, have passed known edits.

The data used as the source for extractions must meet the expectations of the data call in both data elements and time.

Security

Organizations are verified on the network.

Users are able to participate only in the context of their identified organization.

Data Privacy

Raw Insurance Data remains private to the owner and is never visible to any other entity.

Extraction results are controlled and only the designated reporting analytics node can ever see individual results.

Performance

Scalability

Availability

Availability defines when the system is usable. A useful metric is 24X7X365, which means it is always available.

To achieve always up, we must have a self-healing, fully redundant system.

This is achieved differently for different parts of the architecture.

Inside Kubernetes

Outside Kubernetes

Customizable

Member nodes can be adjusted to fit the customers' needs.

The customizations can be privately maintained or shared as open source.

Deployable

Strong automated tests.

CI/CD (Continuous Integration/Continuous Delivery) pipeline that executes the tests.

Observable

Can subscribe to events that occur in the system.

For business reasons.

For technical reasons.

Able to see logs for all parts off the application.

Usability

Maintainability

Easy to contribute.

Easy to test.

Can write automated tests for the APIs, the UIs and any individual projects.

Easy to debug.

Monitoring

Standard technical notifications of issues in the system.

Standard business notifications of issues as they occur.

Logging

Standard error messages and log statements.

Support

How do we provide support for openIDL? How can users report issues? How do we track those issues and ensure resolution?