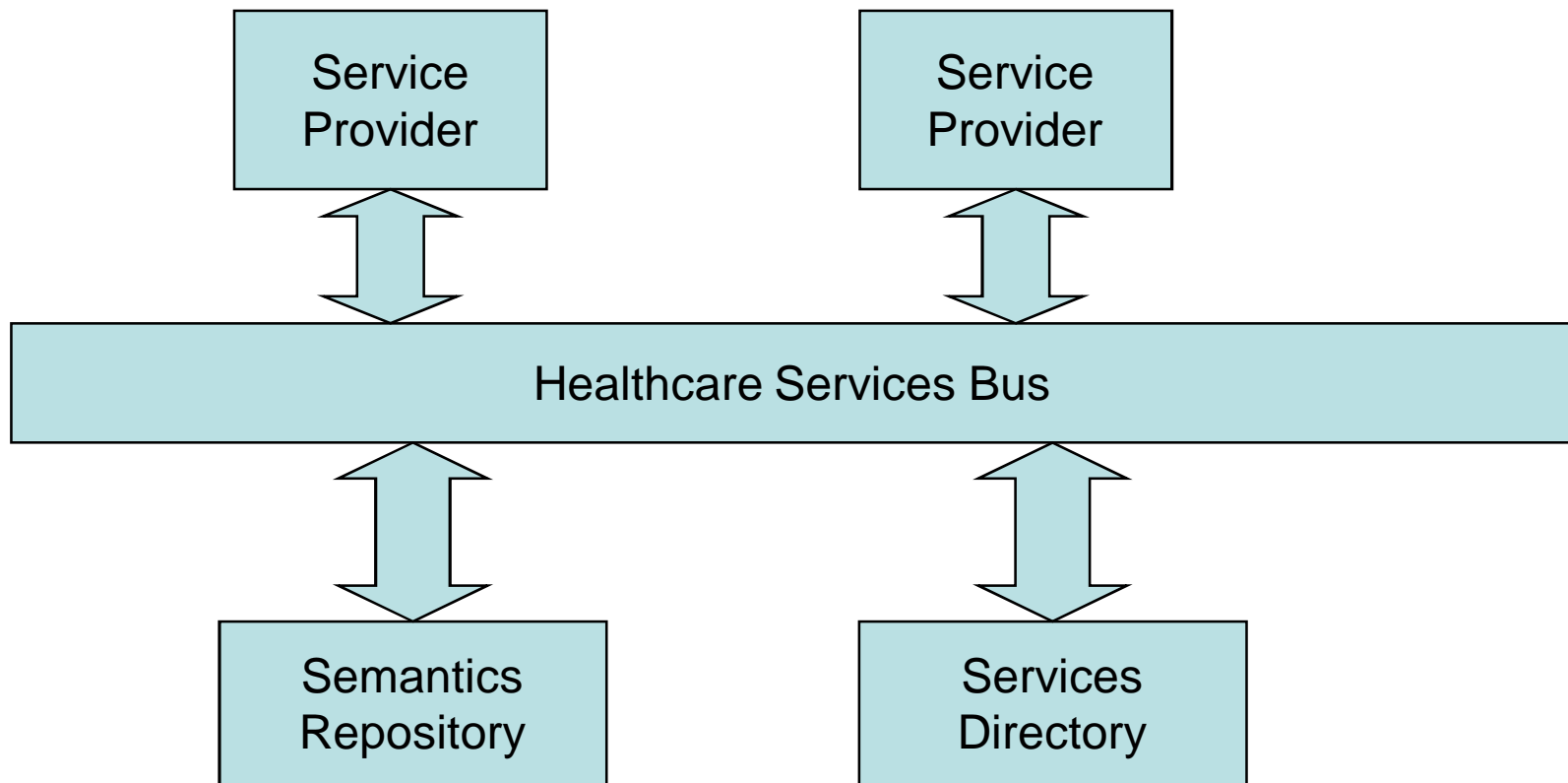


Open Health Tools

Tooling Architectural Vision

Technical Architecture



Semantics Repository

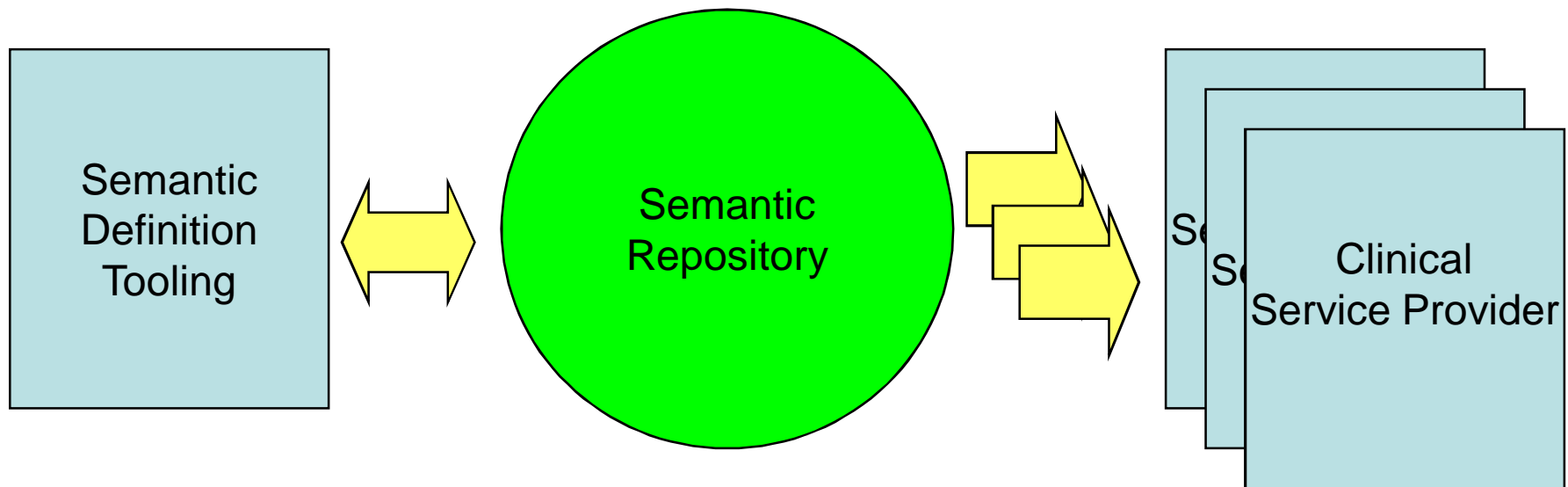
Terminology

Archetypes with Terminology Bindings

Templates

- Archetype restriction and assembly
- UI definition?

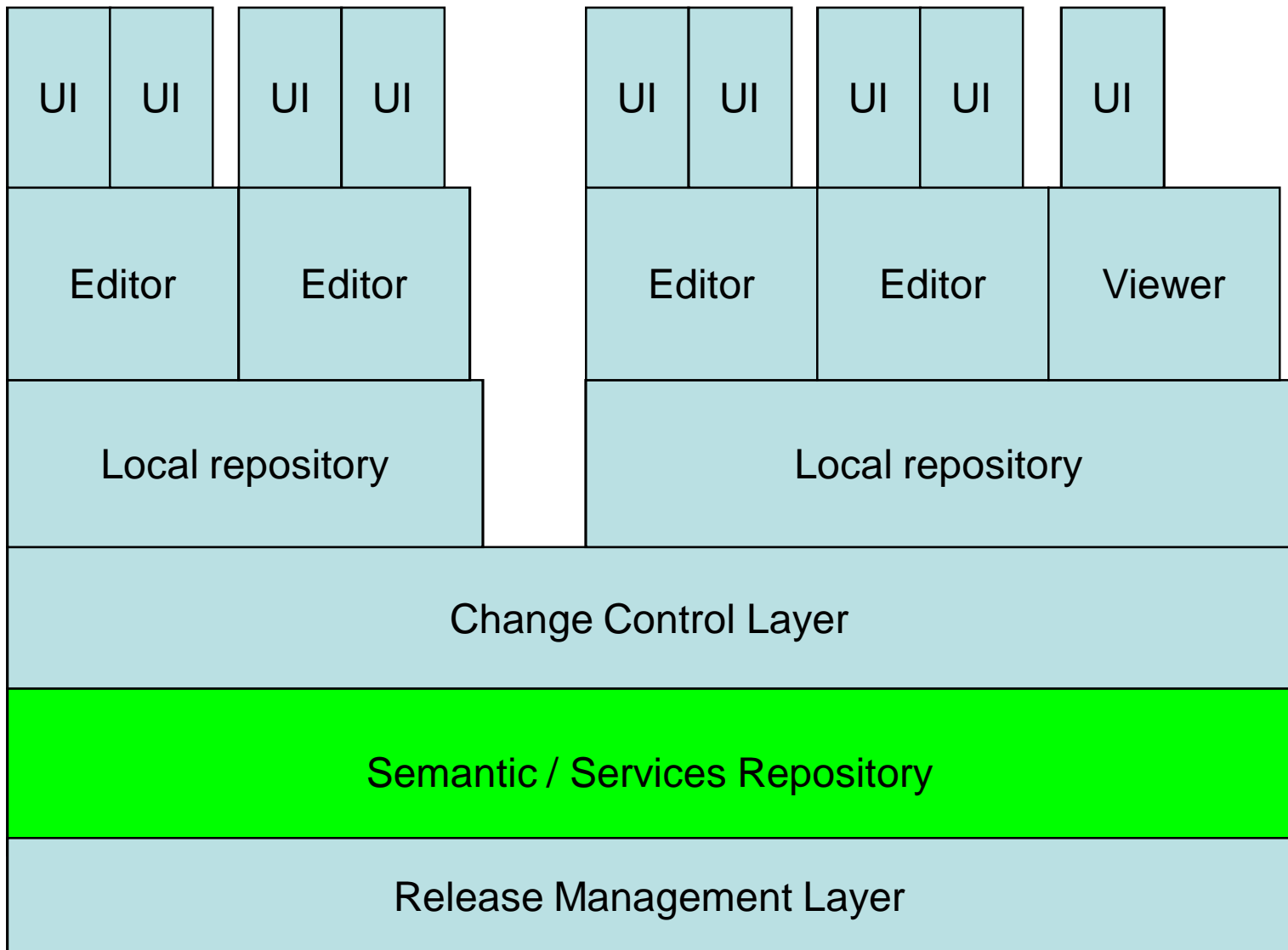
Semantic Definition and Use



Requires good tooling to support:

- Semantic definition and terminology modelling
- Use within the Clinical application

Tooling Architectural Layers



Why a two-level repository architecture?

Different editors have different demands on the repository performance and behaviour

Individual Members may have different priorities, approaches and technology choices

Terminology development requires high performance searching and comparison

Translation development requires highly distributed changes

The Semantic repository (Global and/or national) needs to be updated on a commit basis, not by all intermediate steps

The Semantic repository needs to be general for all editing and upload purposes

Semantic Repository

Agreed standard form for the repository
Must allow for branching and merge

Separate Service Layer:

- Used for upload into systems
- Manage release, versioning etc.

Will be a distributed repository in all but the simplest installations

Services repository

Based on current service-oriented architecture principles

Web-services based, where possible

Using HSSP standards, where possible

Change control layer

Ensures legal and consistent change to the semantic repository

Manages branching and merging

Implements simple syntactic QA

Provides commit and roll-back functionality into repository

Local repository

Optimised for local editing purposes

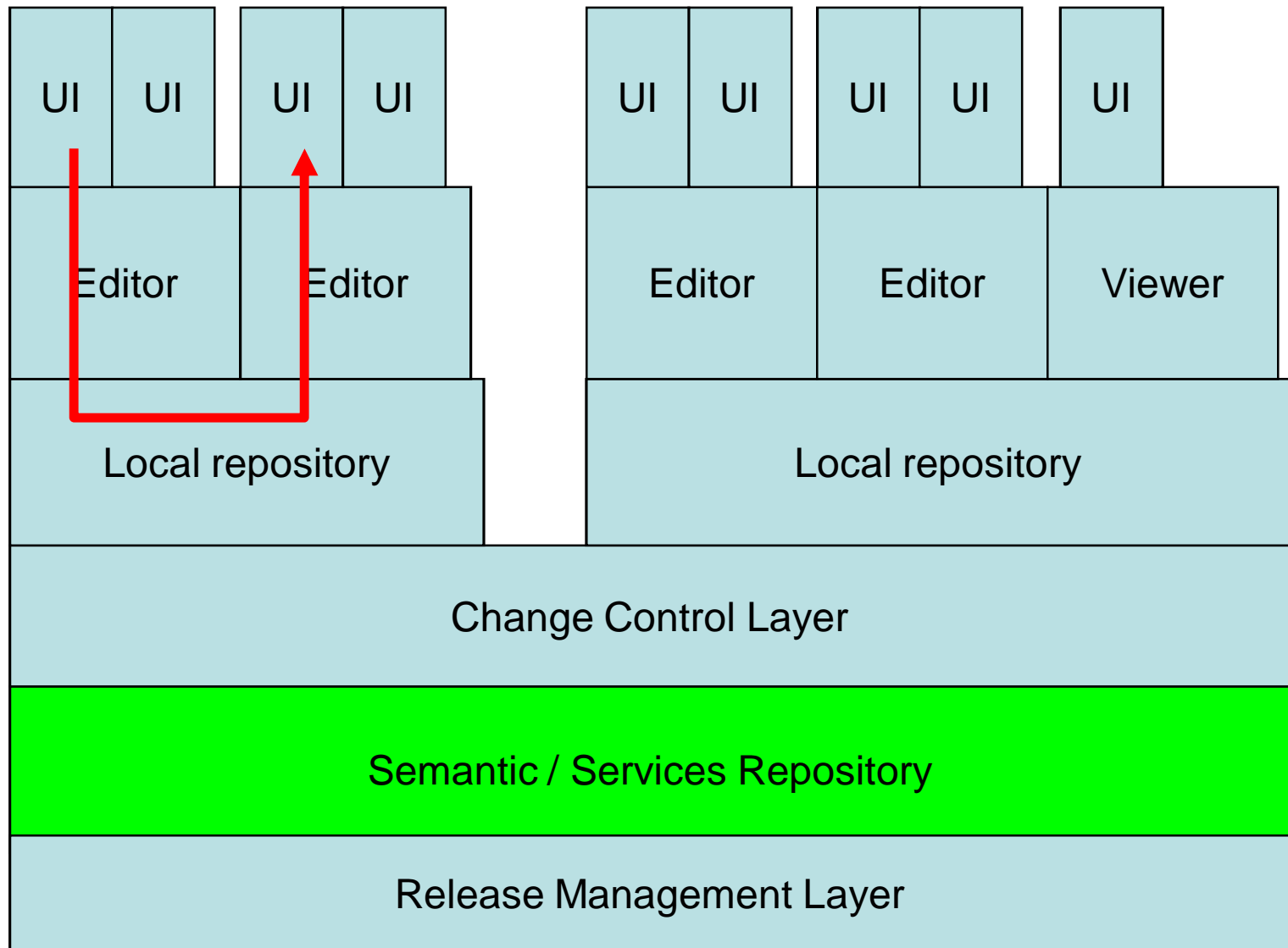
Extracts from the semantic repository through the change control layer

Synchronises changes through the change control layer

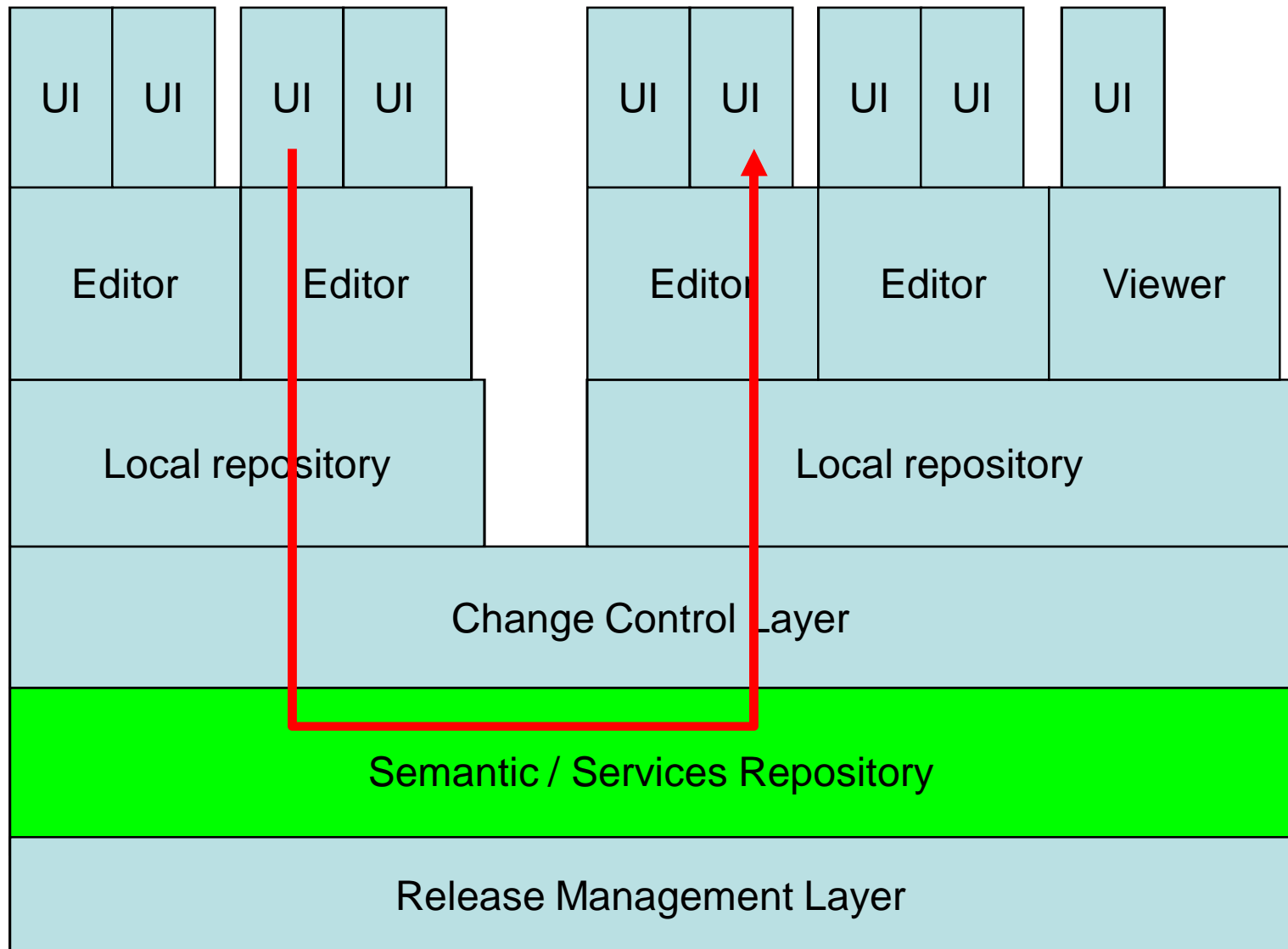
Manages local change control, ensures traceability

Manages local issues tracking, feedback etc.

Localised editing, without commit back to main repository



Editing with commit back to main repository



Conclusion

Tooling should be based on the concept of a shared semantics / services repository

Tools will be developed for the support of standards in a harmonious way

The ultimate tooling stack for interoperability design should be holistic and single platform

The tooling stack today is immature, and a pragmatic and iterative approach is needed to achieve full integration