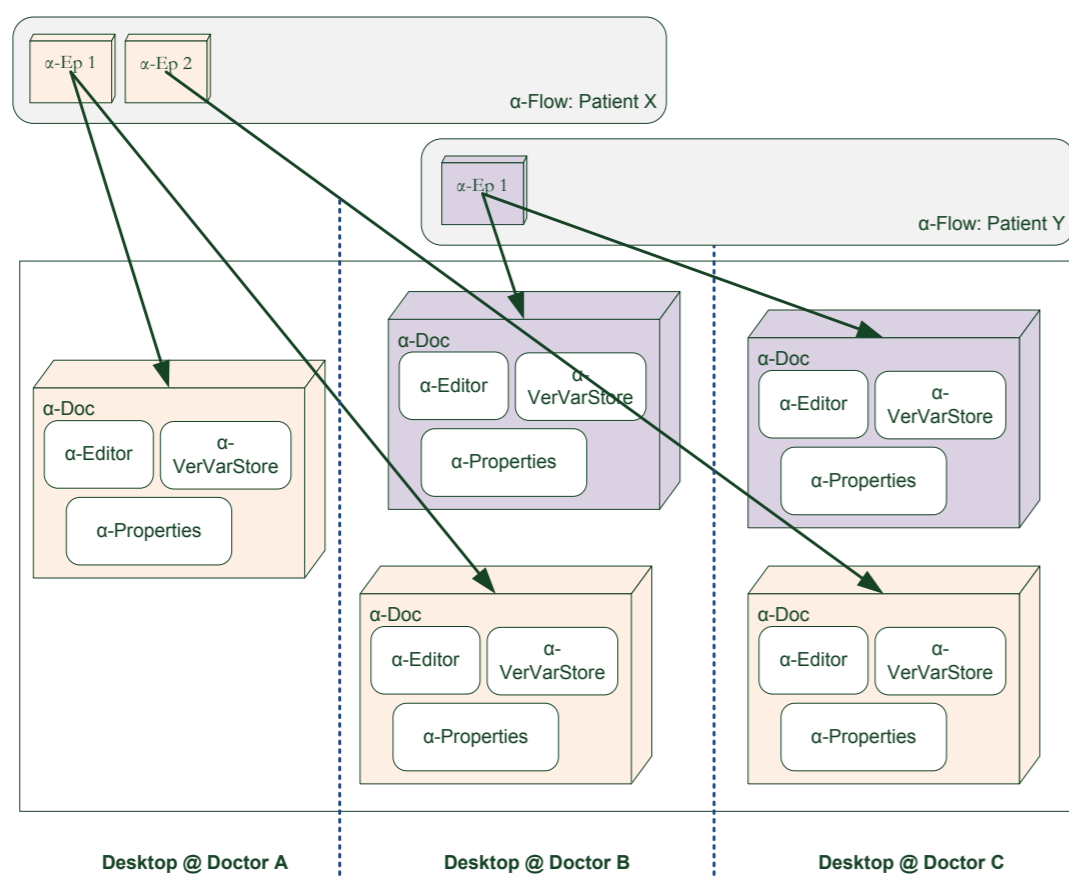


alpha-Props: A Rule-Based Approach to 'Active Properties' for Document-Oriented Process Support in Inter-Institutional Environments

α-Flow

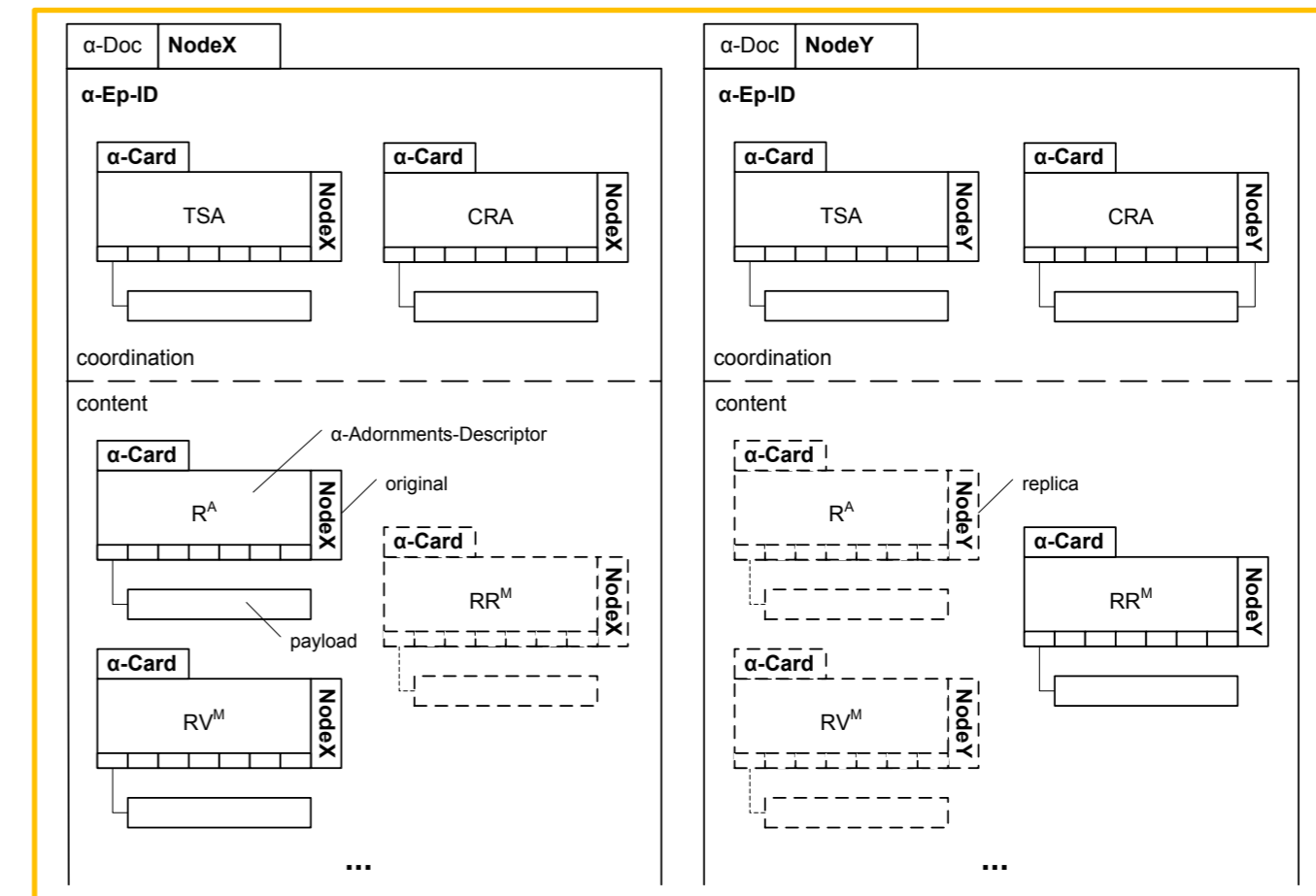
▪ **Idea:** The alpha-Props project aims for the design of a lightweight subsystem, which realizes the active properties in active documents in the context of a distributed and decentralized workflow named alpha-Flow.



Components

- **α-Editor** allows the user to view, manipulate and edit the active document.
- **α-VerVarStore** supports the payload administration of the α-Cards and the storage of the artifacts.
- **α-Props** are present in every α-Doc; can only see and work with the artifacts of this container or its distributed replicas.
- **α-Episode (α-Ep)** is:
 - a distributed process characterized by a particular goal and constructed of a number of distributed activities;
 - represented by one α-Doc.

α-Doc



- An **α-Doc** is a document and an application at the same time.
- α-Docs are assigned with active-properties (α-Props).
- Replicas of the α-Docs reside at the desktop of every participant concerned with the α-Episode, this α-Doc represents.
- **α-Cards** represent passive documents (payloads) enwrapped in an **α-Adornments-Descriptor**.
 - **Coordination α-Cards:** two, mandatory for every α-Doc, equally shared by all participants
 - **TSA** (Treatment Structure Artifact)
 - **CRA** (Collaboration Resource Artifact)
 - **Content α-Cards:** none or many
- **α-Adornments-Descriptor** is a set of adornments defining the payload, e.g.
 - validity, visibility, version, variant, etc..

Requirements

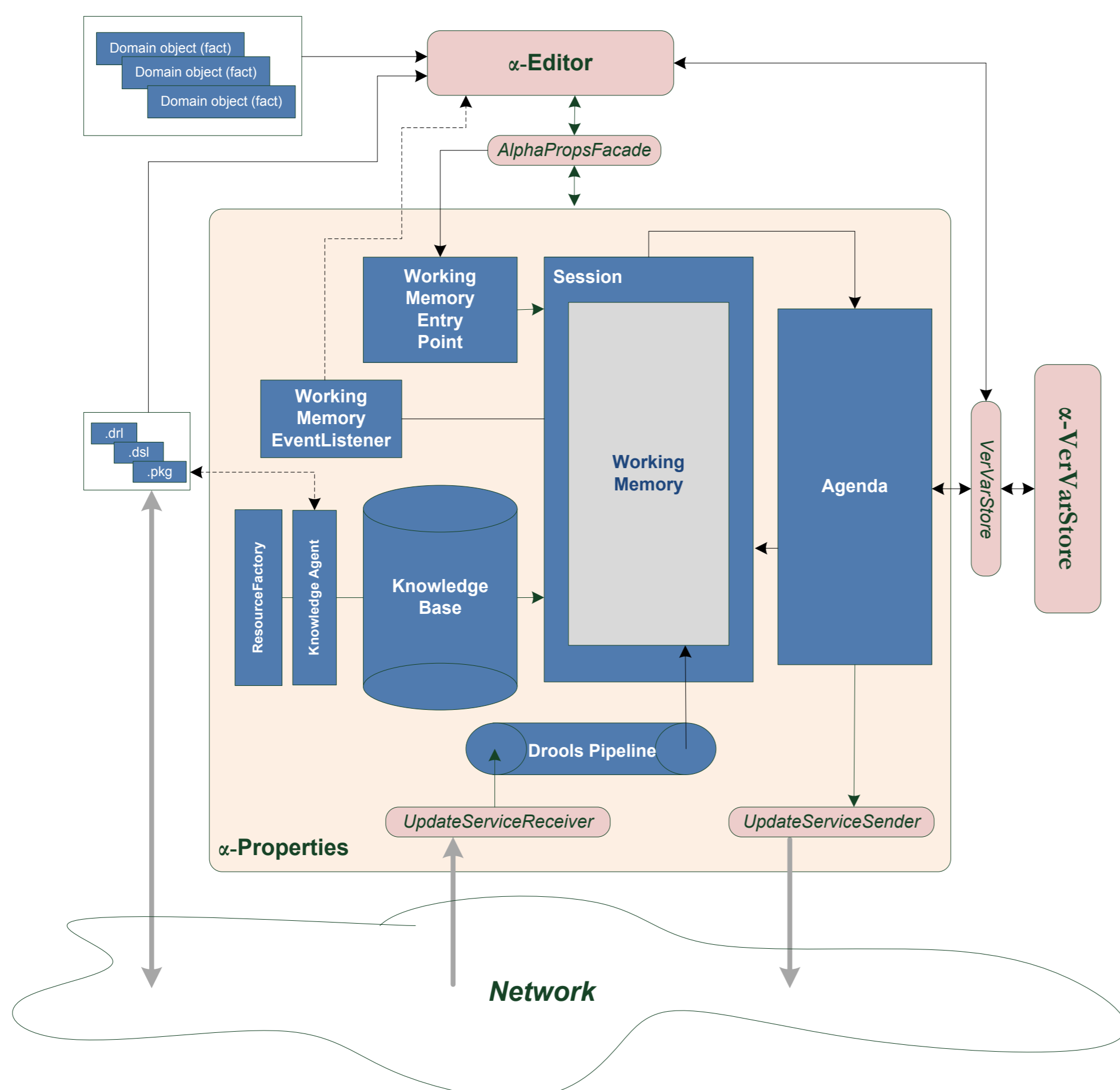
- **α-Flow Requirements Framework**
 - Distribution
 - Autonomy
 - Monitoring of changes
- **α-Props - Requirements of the API**
 - Reasoning over facts, if user triggers event or if some other peer sends an event
 - Managing versions of the artifacts
 - Propagating changes to all involved parties
 - Communicating changes to the Editor
 - Enables Queries over the current state of the artifacts per α-Episode (on fine-grained basis)
 - Providing dynamic upload of rules
 - Light-weight
 - Promptness

The Usage of a Rule Engine

- **Motivation**
 - Rule engines are capable of reasoning over facts and picking up responsive actions based on them.
 - They allow for deploying application-specific rules dynamically and have scalable architecture.
 - Rule engines enable the handling of repetitive tasks.
- **Deployed Technology:** *JBoss Drools*



System Design



Adding an α-Card: Sequence of Events

