

DECADE Architecture

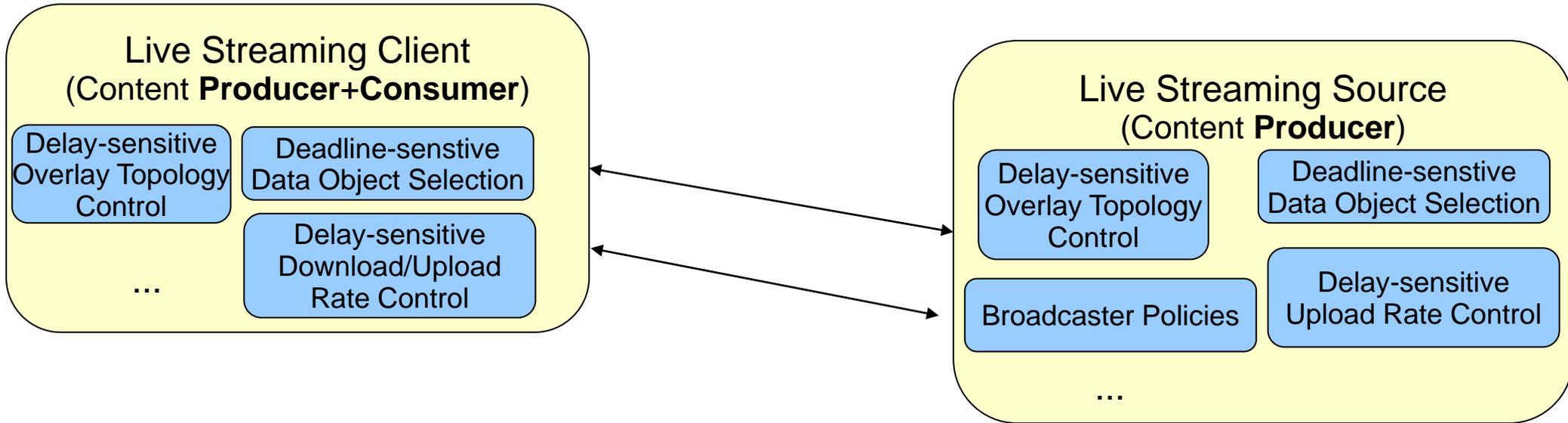
draft-alimi-decade-arch-01

Richard Alimi, Lijiang Chen, Hongqiang Liu,
Dirk Kutscher, Akbar Rahman, Richard Yang

Draft Purpose

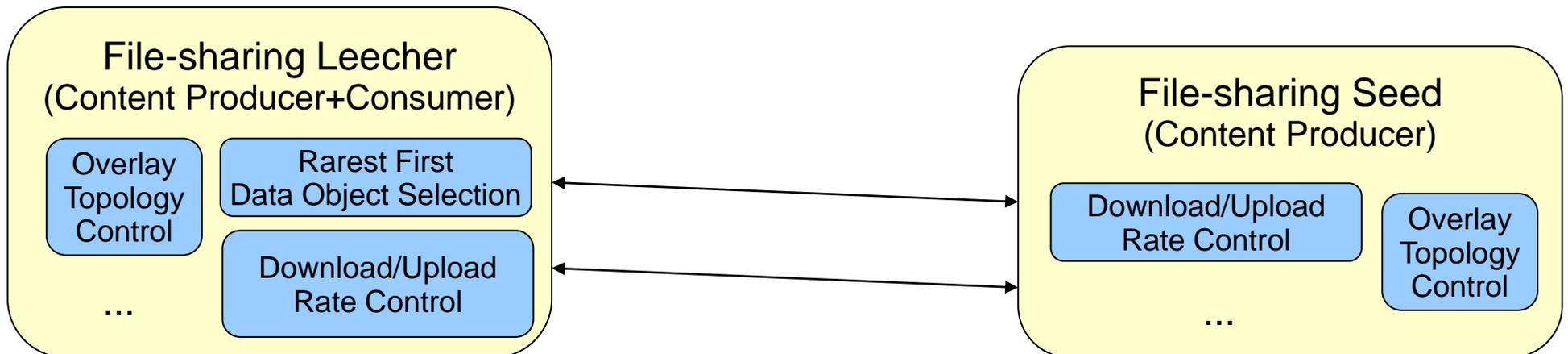
- DECADE intended to improve network efficiency of P2P apps
 - .. by introducing storage into the network
- Draft presents one possible architecture for DECADE
 - Satisfies requirements from draft-ietf-decade-reqs
 - Benefited substantially from draft-ietf-decade-survey
- Focus on major design issues
 - Describe and motivate major design principles
 - Define remaining issues to be resolved
- Feedback much appreciated

Architectural Entities: Applications

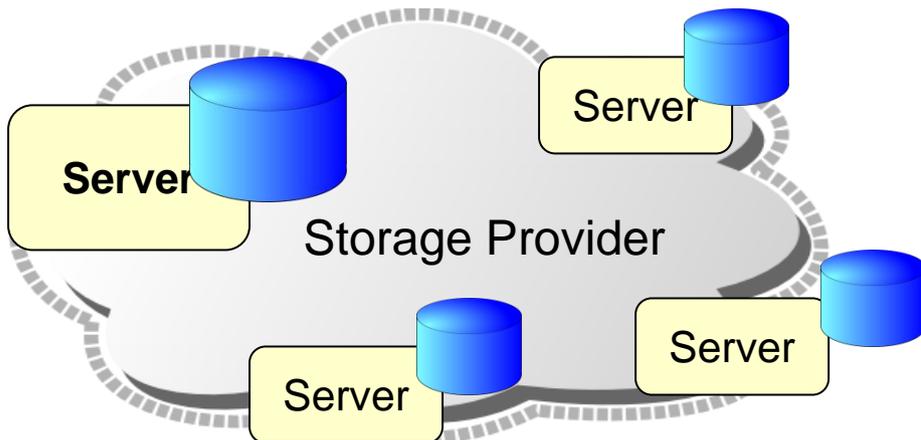
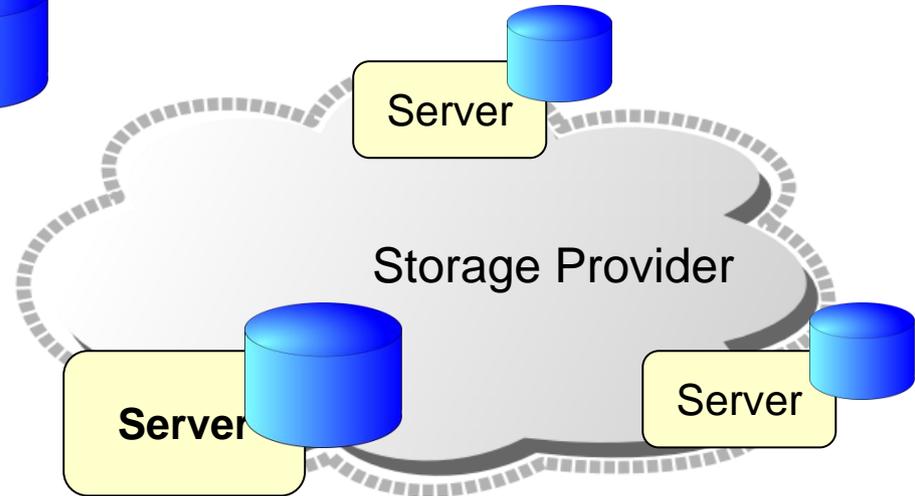
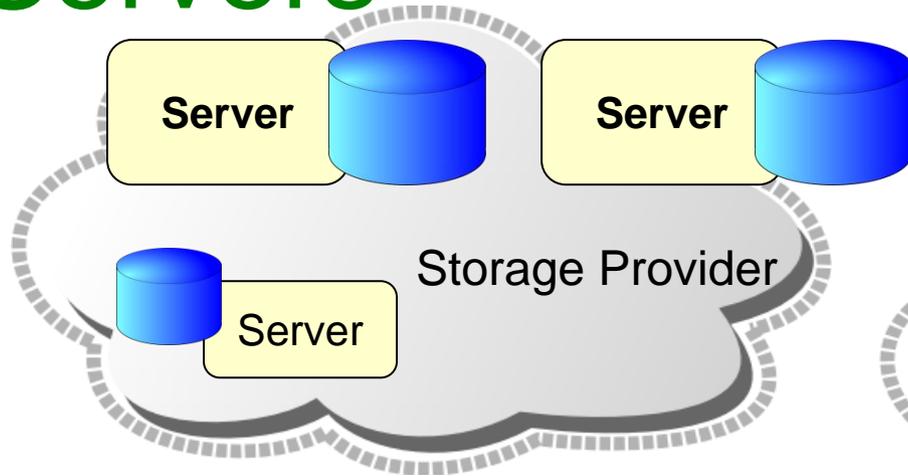


Typical Characteristics:

- Divide content into smaller objects for distribution***

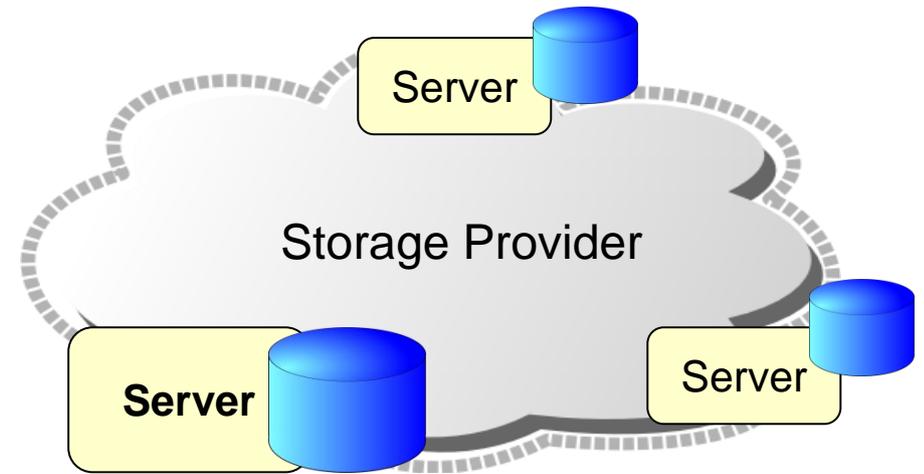
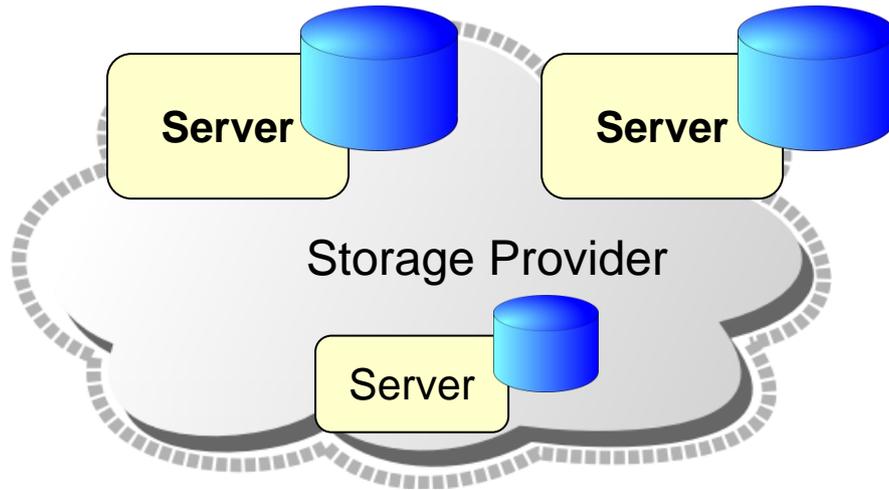


Architectural Entities: DECADE Servers



- Deployed across independent storage providers
- Servers (typically) are **upstream** of last-mile
- Number of servers, resources, placement decided by each storage provider
- Simple servers
 - No standardized distributed coordination

Architectural Entities



Many-to-many relationship

- *Single DECADE server used by multiple application producers/consumers*
- *Single application producer/consumer may use multiple DECADE servers*

Live Streaming Client
(DECADE Producer+Consumer)

Live Streaming Source
(DECADE Producer)

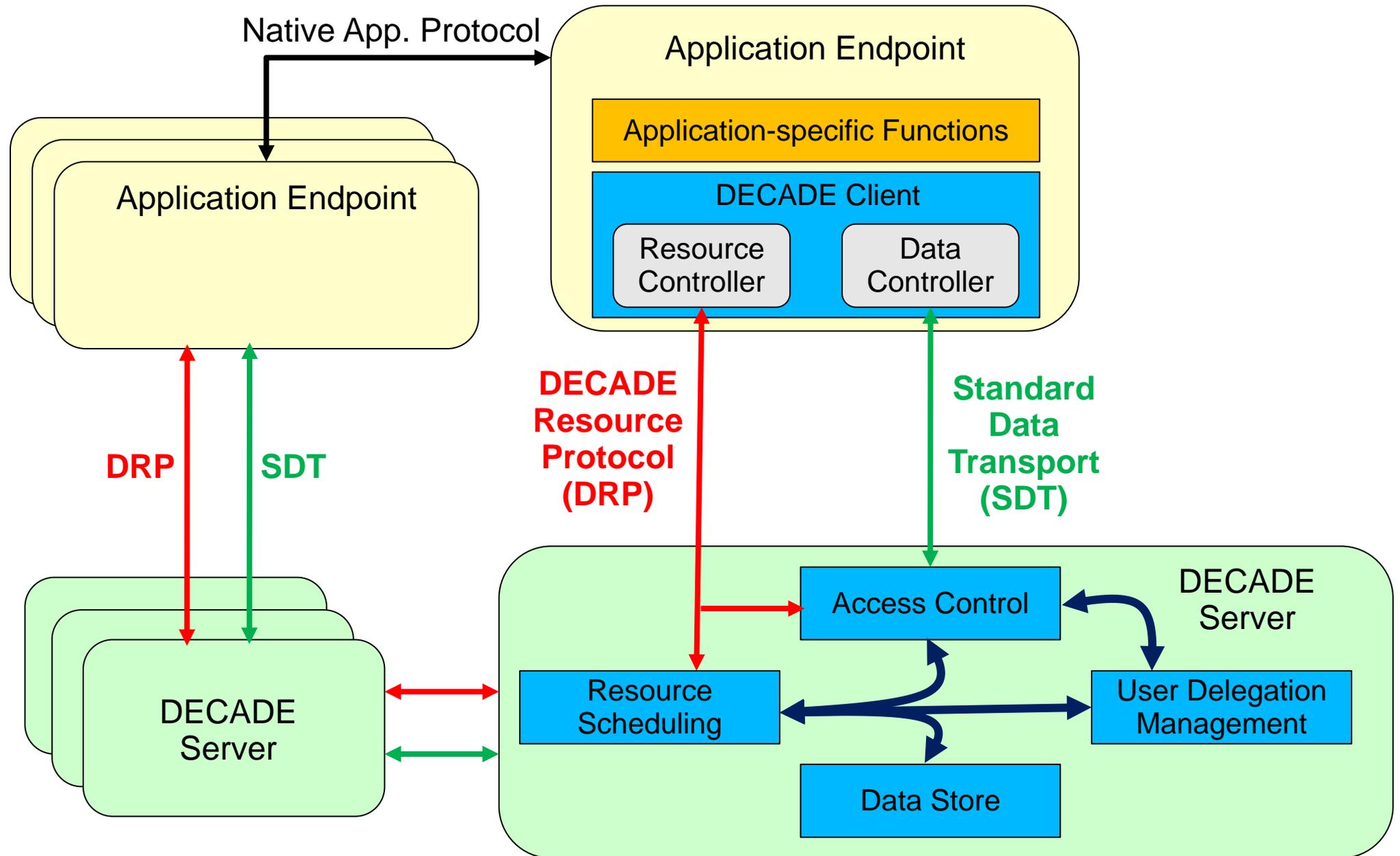
File-sharing Leecher
(DECADE Producer+Consumer)

File-sharing Seed
(DECADE Producer)

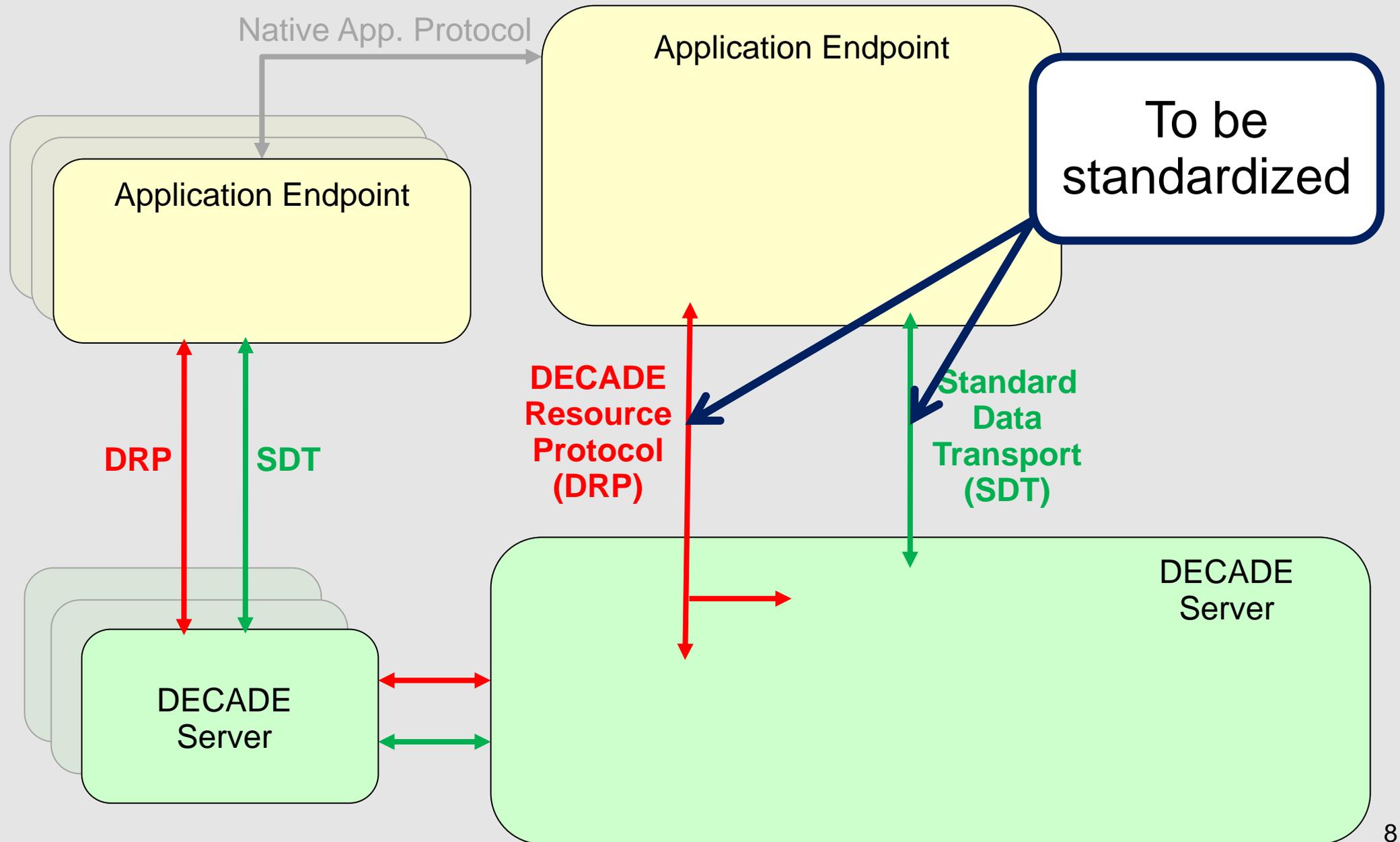
DECADE Architecture Goals

- Does NOT attempt to directly implement application policies
 - But, provides necessary primitives for applications to implement policies when controlling data transfers amongst in-network storage elements
- Identify DECADE's relationship to existing IETF protocols
 - ... and where (if any) new protocol is needed or extensions to existing protocols need to be made.

Decoupled Control and Data Planes



Decoupled Control and Data Planes



Architectural Principles

Architecture has a core set of underlying principles

- Decoupled control and data (already discussed)

- Immutable data objects

- Decoupled naming

- Explicit control

- User delegation

Data Model and Naming

- Immutable data objects
 - Common amongst content distribution applications
 - Reduce consistency requirements (e.g., no need to lock across updates)
- Separation of location and identifier
 - Application-level names still used
 - Application remains in control of data location
 - Application-level index maps App names to DECADE names
 - DECADE does not need to provide distributed coordination/indexing amongst servers
- Naming
 - Discussion at end of presentation

Explicit Control

- Applications have explicit control over
 - Stored data
 - Store, retrieve, delete (upon request or via TTL)
 - Data location
 - Choose DECADE Server to store data
 - Data transfer path
 - ... amongst application endpoints and DECADE servers
 - Access and resource control policy
 - ... for server resources consumed by application endpoints

User Delegation

- Motivation
 - Applications share physical resources
 - Implies need for allocation/sharing policy
 - Not scalable for provider to directly manage app-level access and resource control
- Delegate resources via *users*
 - Provider grants resources at DECADE server to a *user*
 - User may be: end-user, content provider, application provider
 - User then allocates granted resources to one or more applications
- Protocols for user management NOT standardized
 - Assume this is done out-of-band

Map Survey Criteria to Architecture

- Data Access Interface
 - Read and write objects via DECADE Client's Data Controller
- Data Management Operations
 - Move or delete objects via DECADE Client's Data Controller
- Data Search Capability
 - Implemented by application (e.g., local lookup or distributed index); may consult local data index in the DECADE Client's Data Controller.
- Storage Mode
 - Immutable objects; implemented by storage system underlying the DECADE server (underlying implementation may or may not be object-based).

Map Survey Criteria to Architecture

- Access Control Authorization
 - Policies generated by application. DECADE Server responsible for implementing access control checks.
- Resource Control Interface
 - Policies generated by application. DECADE Server is responsible for implementing the resource sharing policies.
- Discovery Mechanism
 - Outside scope of the scope of *current* DECADE architecture document.

Discussion: Data Object Naming

Problem Space

- Name is function of content
 - Example: hash
- Else
 - Structure of name
 - Flat (e.g., UUID), hierarchical (e.g., use namespaces)
 - Who generates?
 - Client or server?
 - Has effects on certain optimizations that can be made

Discussion: Resource Control Model

- How are resources delegated?
 - Current model has user delegations: Server → User → App
- Which resources?
 - Network bandwidth, disk I/O, memory, CPU?
- How are resources controlled?
- ***Objective***
 - Ideally leave as much to implementation as possible

Feedback and Next Steps

- Comments on ...
 - Architectural principles?
 - Logical protocol separation?
 - Resource/access control and data transport
 - Identified discussion items
- Next Steps
 - Include decision on naming and resource control
 - Include discovery component