

DECADE Example Solution Architectures

Richard Alimi
Richard Yang
Haibin Song

Overview

- Focus of Architecture WG work item:
 - Identify relationship with existing protocols
 - Identify where new protocol is needed and what functions needed

- What can DECADE look like?
 - Chartering has purposefully stayed away from solution proposals

- ***Purpose of this Presentation***
 - ***Provide examples of possible solution architectures***
 - ***Give a flavor for what DECADE can look like***
 - ***NOT meant to be restrictive***

Functional Components

■ DECADE Server

- ❑ Manage user accounts (and resource sharing policies)
- ❑ Manage users' apps (and resource sharing policies)

■ Storage Server (may be co-located with DECADE Server)

- ❑ Handle read/write/delete/etc requests

■ DECADE Client

- ❑ Define and update user's apps (and resource sharing policies)

■ Storage Client (may be co-located with DECADE Client)

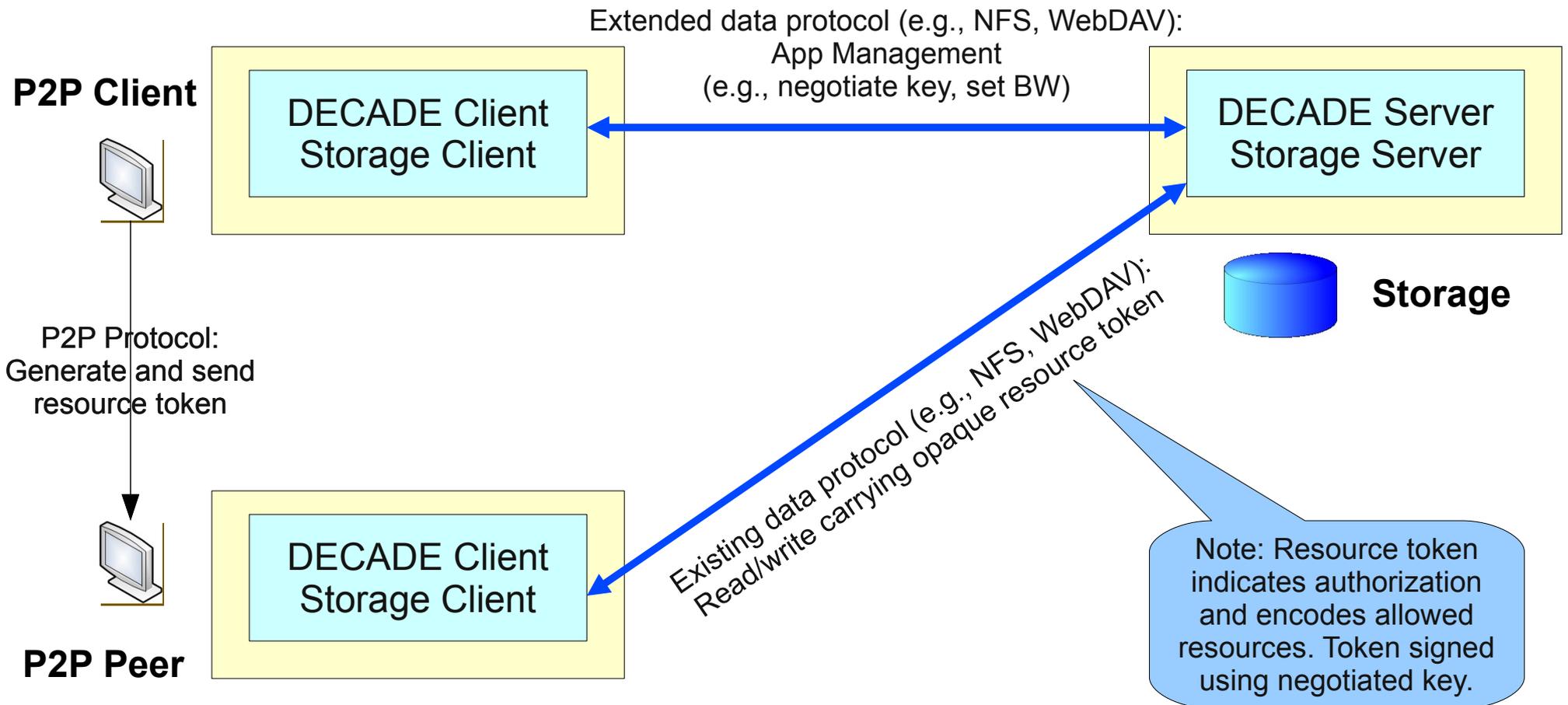
- ❑ Send read/write/delete/etc requests

System Assumptions

- *Following architectures make certain assumptions*
 - *Meant to illustrate some concrete design approaches*
 - *May change during course of WG*
- A1: Resource Allocation Model
 - Storage provider allocates resources amongst one or more users
 - User allocates resources amongst one or more applications
- A2: Replication
 - Not provided by storage provider
- A3: AuthN/AuthZ for remote peers
 - Explicit separation of “client” and “resource owner” (a la OAuth)
 - But, tokens generated and distributed by P2P protocol by owner

Architecture 1

- Directly extend data transport protocol(s) with new operations
 - Would be done for each data transport wishing to use DECADE



Architecture 2

- New protocol managing user/application resource parameters
 - “Exports” resource control parameters to a data transport
 - Similar to IKE workflow

