

Deployment Patterns Draft Update

<http://tools.ietf.org/id/draft-levine-vwrap-deploy-01.txt>

David W. Levine

dwl@us.ibm.com

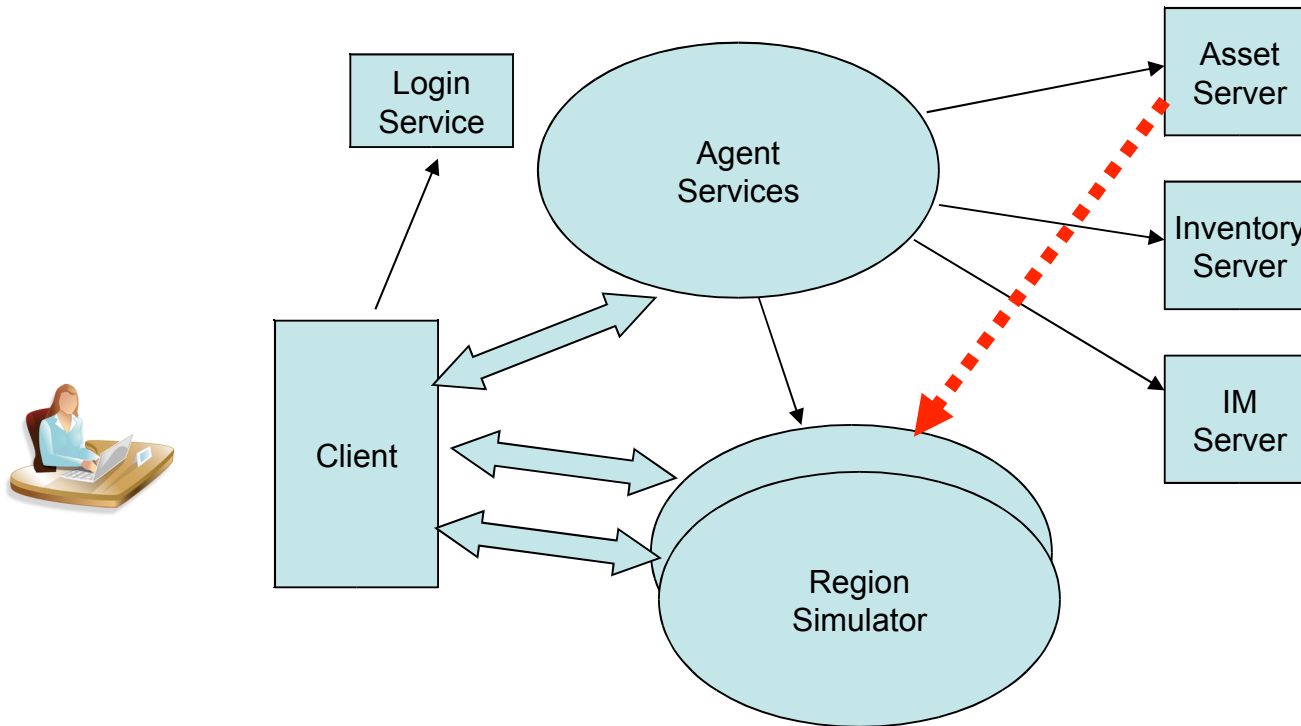
IBM Research

IETF 77 March 23, 2010

Overview

- Topics not currently in the intro / foundation
- Probably ready to roll in portions during next editing cycle – Need to get serious about incorporating new material
- Focus on making sure we capture the ways the underlying stuff is being deployed
- Related to Client Side Caps
- Asset Content Distribution material is new

Adding an Agent Domain



- Effectively adds a second facade for agent oriented services
- Breaks apart the monolithic trust cloud into multiple (but very statically defined) groupings
- Second asynchronous event pipe
- Routing is still very implicit
- “Addressing by region handle” fails to fully work
- The Region / Asset server line represents a fault line

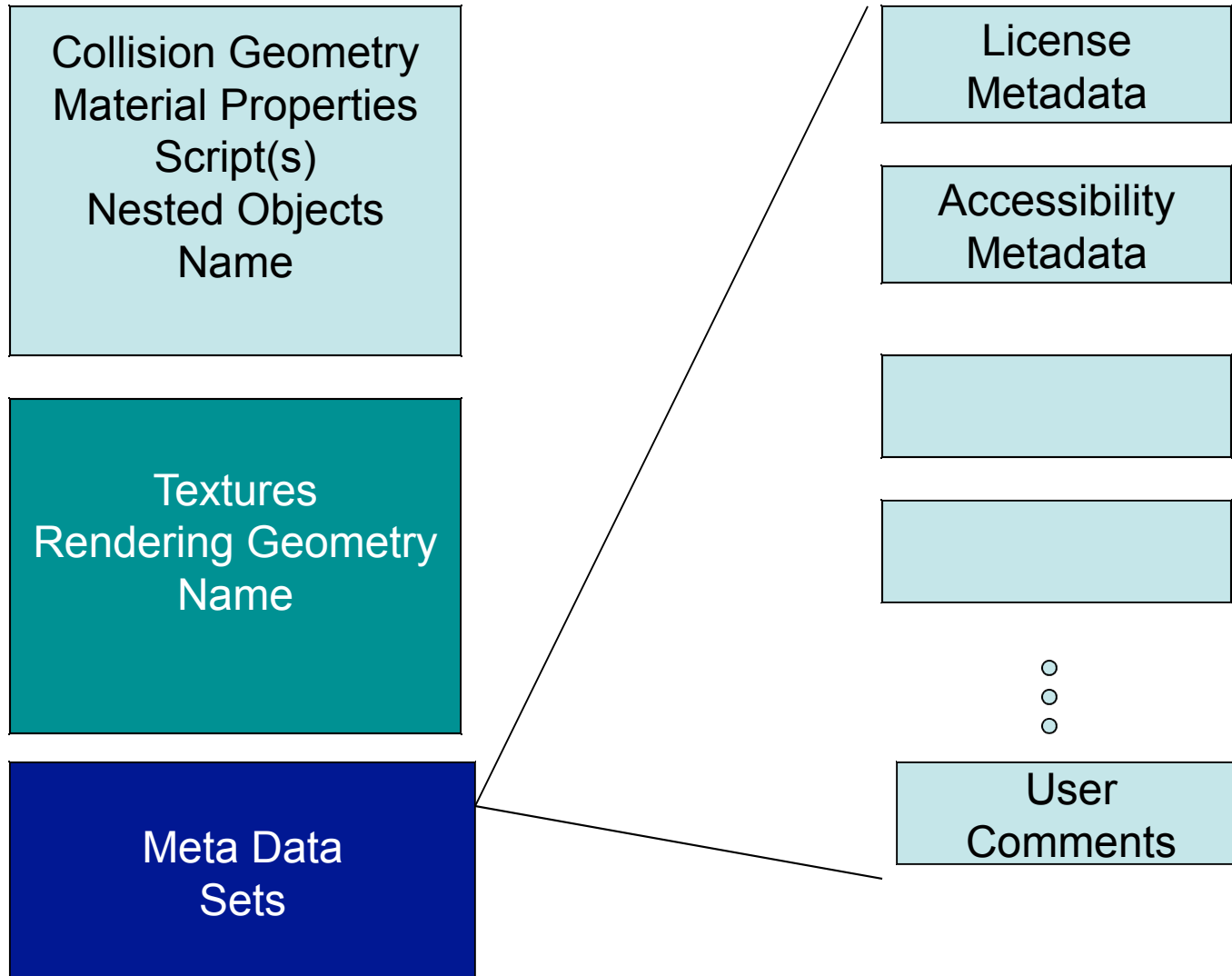
Fault Line?

- Exposing Assets to regions across Trust / Admin boundaries
- When we allow regions and asset servers to interact across “domains” things change
- We are creating new public interfaces
- This is fundamental to meaningful interoperation
- More in Client Side Caps

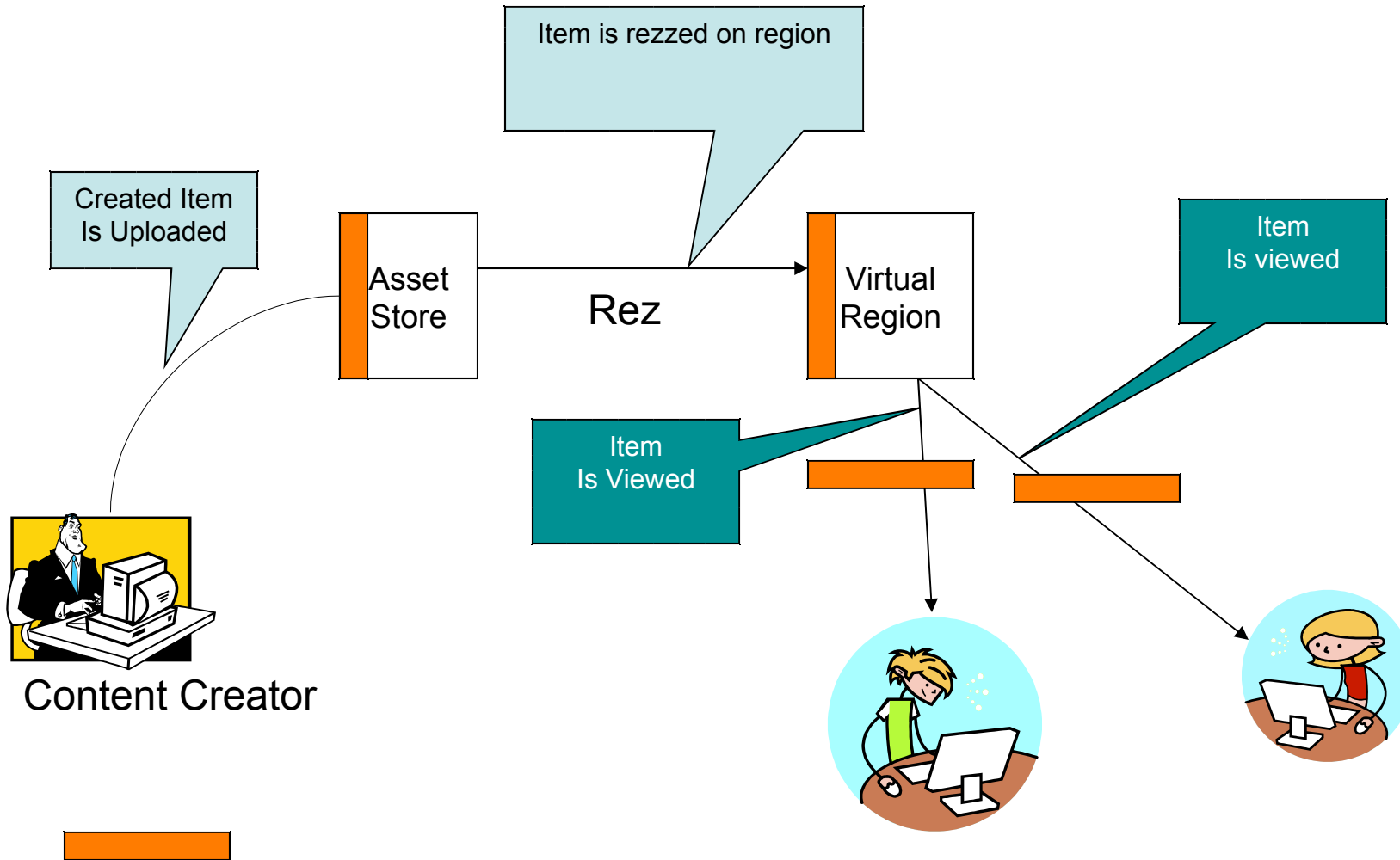
Asset

- We may need a better name
- Used in several roles:
 - Placed into a virtual space (rezzed)
 - Attached to an avatar
 - In a server waiting to be used
- Has multiple parts
 - Simulation Component
 - Representation Component
 - Meta-Data
- Each part may have multiple representations
 - Levels of Detail
 - Licensed vs Freeware copies
 - Primset vs. Sculpty vs. Mesh
- Always get either the content or a URI to fetch the content at each level in description

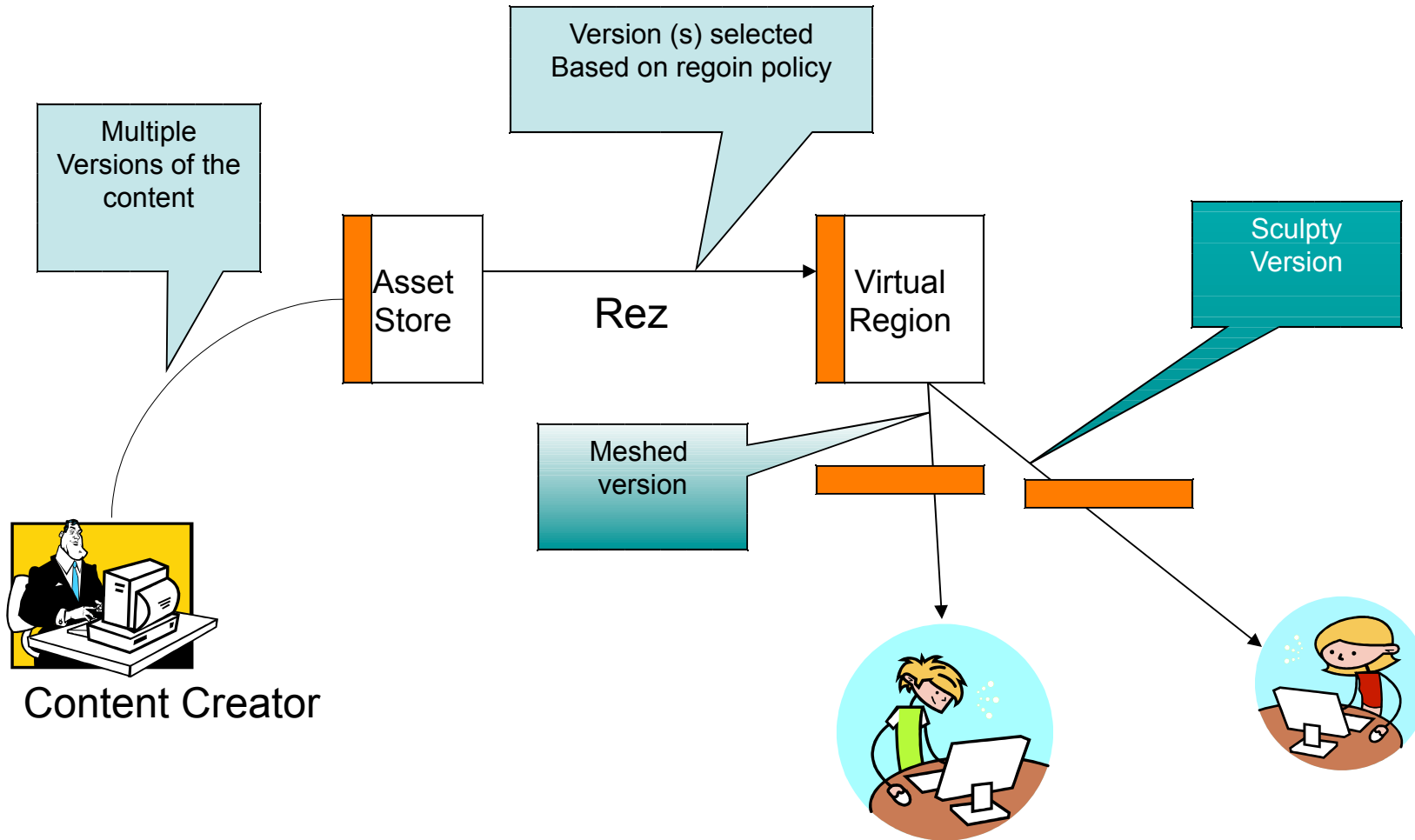
Asset: Visually



Content Creation/Consumption flow

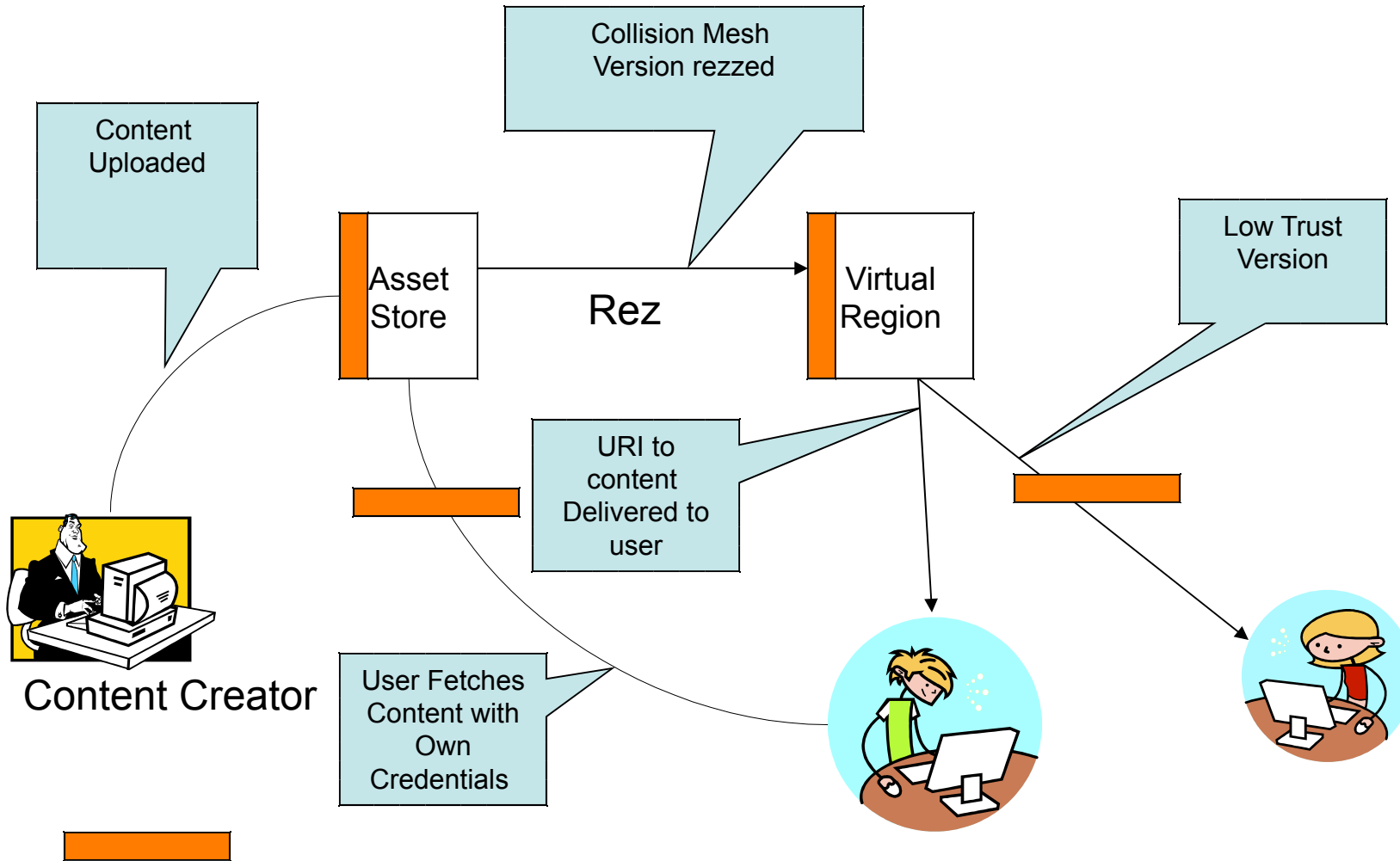


Alternate content flows



- Trust, Client support of content type, license.

Indirect Content Access



Getting Assets Right

- Creates a cleaner model
- Inserts leverage points for key new features
 - Meta-data about use terms for assets
 - Meta-data for accessibility
- Multiple representations
 - Opens the door for flexibility in deploying new content types
 - Allows a range of policy choices for deployers when adding new features
 - Follows the web pattern
- Data vs URI
 - Follows web pattern
 - Allows a range of security / control points
 - Permits very light weight simulated spaces for some use cases

Blob / URI

- The protocols should allow us to include either the “data” that is being conveyed or a URI at every plausible slot in the messages
- Some URIs/URLs actually are service points
 - Can’t inline a “cap” or “service endpoint”
 - Is there any reason why everything else can’t be indirected (Within bounds of good performance)