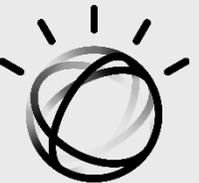


OSLC Linking profiles

Eran Gery

Global Industry Solutions Lead – IBM ELM

OSLC open project member & evangelist



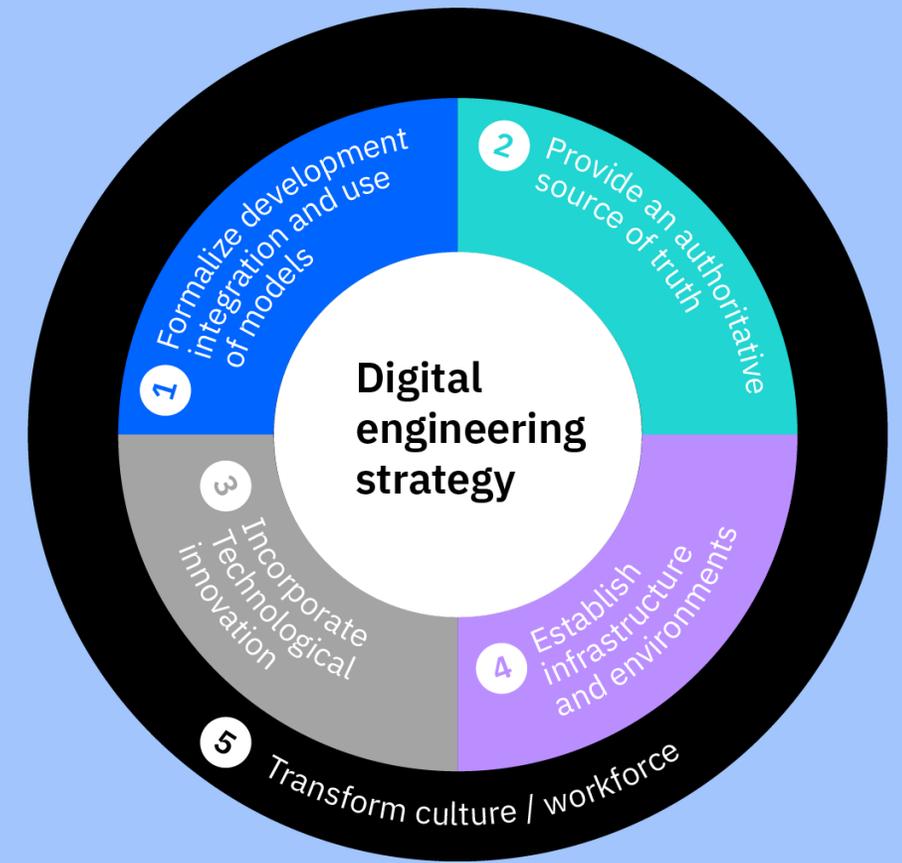
Content

- Digital threads
- What are OSLC profiles
- OSLC linking profiles
- Linking use cases and interoperability goals
- The key aspects of linking profiles
- Related Future Work

Industry vision: Digital threads

Digital threads is "a data-driven architecture that links together information generated from across the product lifecycle and is envisioned to be the primary or authoritative data and communication platform for a company's products at any instance of time."

Digital threads architecture supports the broader vision of digital engineering

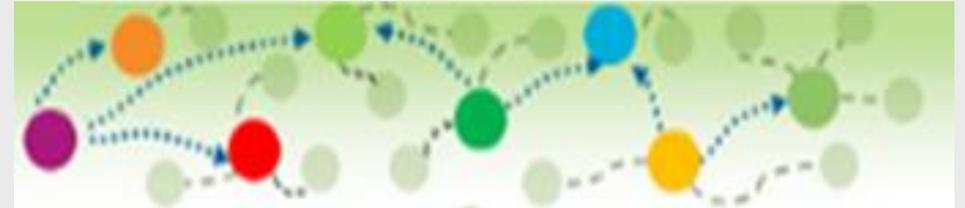


"...such engineering environments will allow DoD and industry partners to evolve designs at conceptual phase, reducing the need for expensive mockups, premature design lock, and physical testing."¹

Key digital threads enabling capabilities

Ensuring cross lifecycle data integrity and stakeholder insights

- **Digital continuity:** establish digital relationships across all engineering data artifacts across all tools and domains
- **Enable cross domain data exchange:** through standard domains vocabularies (ontologies)
- **Global configuration management:** manage consistency across all engineering data sources using cross tools configuration management
- **Cross lifecycle analytics and reporting:** produce the necessary insights and evidence based on concrete engineering data
- **Integrated change and process management:** across all engineering data and tools



OSLC profiles - background

- OSLC specifications allow for “recommended” or “optional” clauses, resulting in vendor interpretations that result in interoperability challenges
 - Practically requires “peer2peer” testing for any two clients and most likely tweaking the implementation
- OSLC profiles are non-normative supplement to OSLC specifications, aim to ensure interoperability between OSLC enabled applications specific use cases specified by the profile
 - Profiles are specified for specific interoperability use cases
- Profiles eliminate the variability of the specifications by requiring mandatory clauses to ensure interoperability of the use cases
- Profiles also provide additional guidance and examples to help implementors produce interoperable implementations effectively
- The creation of the profiles is based on vendors and integrators practical experiences with interoperability challenges

OSLC linking profiles

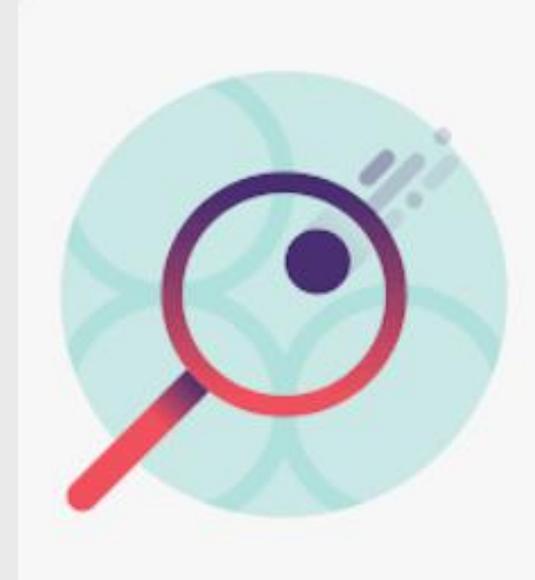
- Focus on the most fundamental linked data use case: establishing digital threads across providers using OSLC links
- We look at 4 levels of linking interoperability
- BASIC
 - provider-1 establishes a link to provider-2
 - Establishing connection across providers, selecting a target resource using selection dialogs, persisting the link for future navigation or resource preview
- Bi-directional
 - provider-1 establishes a link to provider-2, and both providers are “aware” to the link and utilize it
 - Normalizing link storage and discovery of the incoming link
- Configuration aware linking
 - A bi-directional linking in context of OSLC GCM enabled providers
 - Using configuration contexts to select target resources, and to navigate or preview linked resources
- Optimized linking (Future)
 - Optimized bi-directional linking using a link discovery service
 - Utilizing a link discovery service for optimizing incoming links presentation and navigation for incoming clients

Key interoperability aspects of OSLC link profiles

Capability support	OSLC Linking Profile			
	Basic	Bidirectional	Config	Optimized*
HTTPS	MUST	MUST	MUST	MUST
Root Services document	MUST	MUST	MUST	MUST
Authentication	Must	Must	Must	Must
CSP for friends	MUST	MUST	MUST	MUST
CORS for Friends	SHOULD	SHOULD	SHOULD	MUST
Preview Dialogs	SHOULD	SHOULD	SHOULD	SHOULD
PUT on Resources		MUST	MUST	MUST
OSLC Query 3.0 OS support		MUST	MUST	MUST
Store links on the "canonical" side		SHOULD	MUST	MUST
Config Mgmt 1.0 support			MUST	MUST
Contribute links to TRS			MUST*	MUST*
OSLC Link Discovery service				MUST

Root services

- A convention used by ELM and other providers to discover as root for discovery
 - “discovery of discovery”
 - Discover Service provider catalog
 - Also discover other providers such as TRS, Oauth
- Currently not part of OSLC spec
- Linking profile will specify the format of ELM’s root services document



Security related aspects

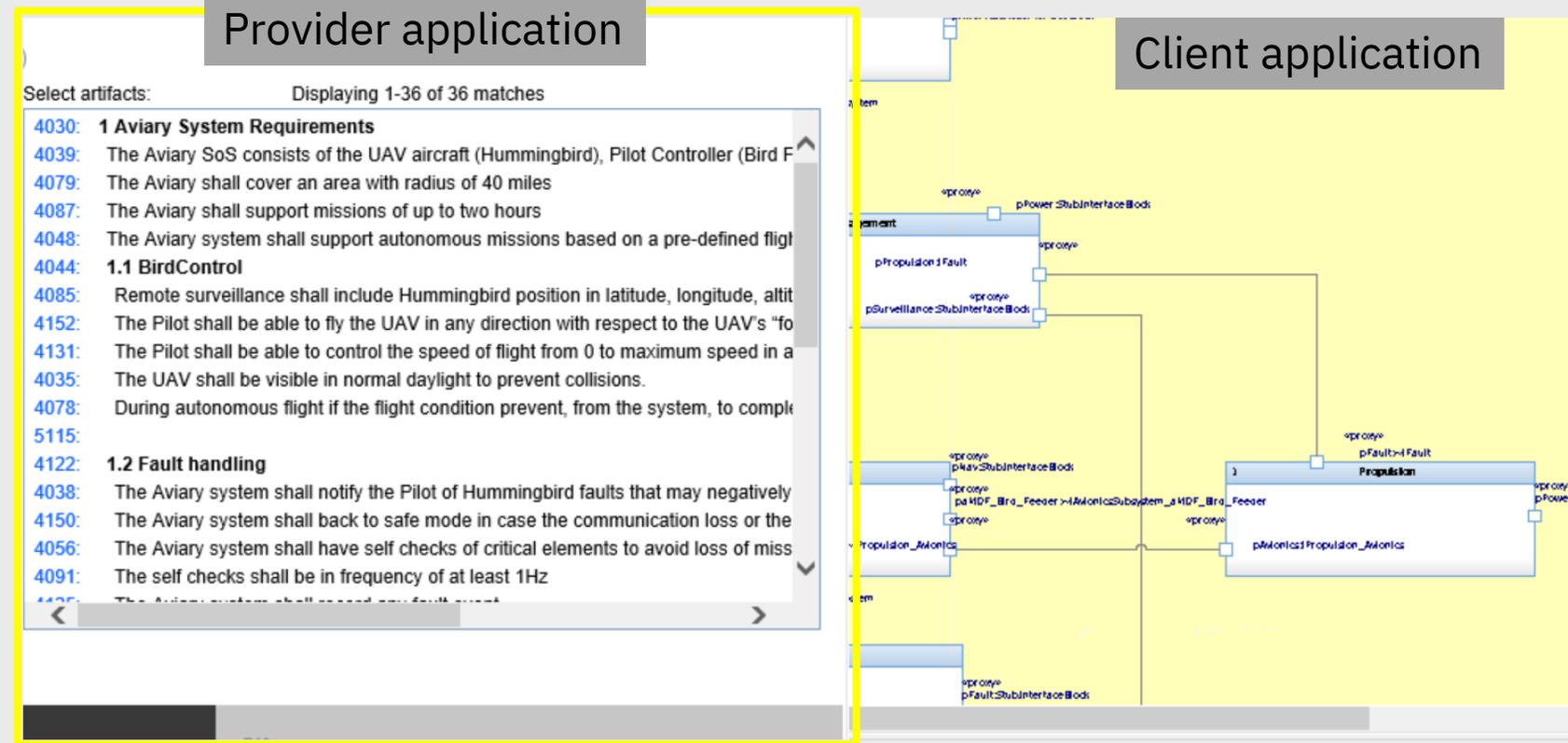
- HTTPS
 - Mandating usage of HTTPS with oslc providers
- Authentication
 - Eliminating the variability related to accessing an OSLC provider. Usage of Oauth1.0, OIDC etc.
 - Required to enable any interaction across providers
- Content security policy (CSP)
 - Currently not mentioned in the spec.
 - Practically required to allow usage of UI (iFrames) inside hosting browsers
 - E.g. previews and selection dialogs
- CROS Origin Resource Sharing (CORS)
 - Currently not mandated
 - Required to enable UI delegation (Javascript)



OSLC selection dialogs

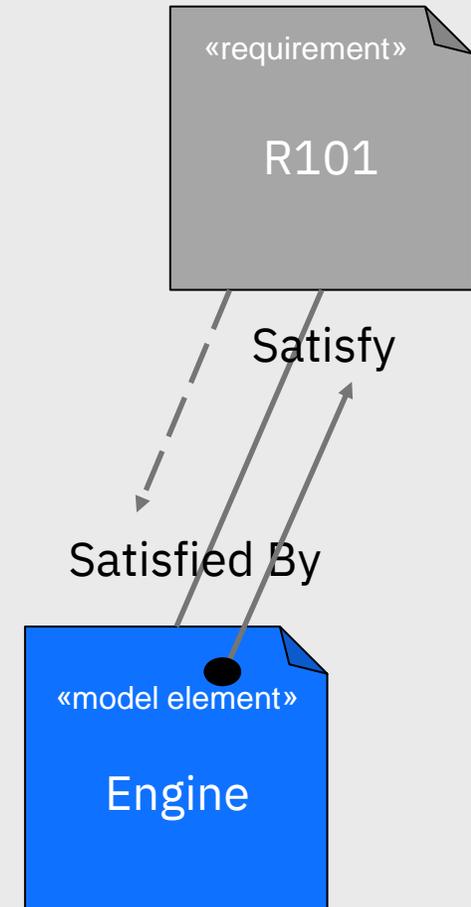
- Delegated selection dialogs are the standard way to link across applications
- The delegated UI uses a presentation of the artifact similar to the native application
- Preferred over query based resource discovery which does not leverage presentation “knowledge” of the trusted application

Delegated requirements application UI for linking modeling resources



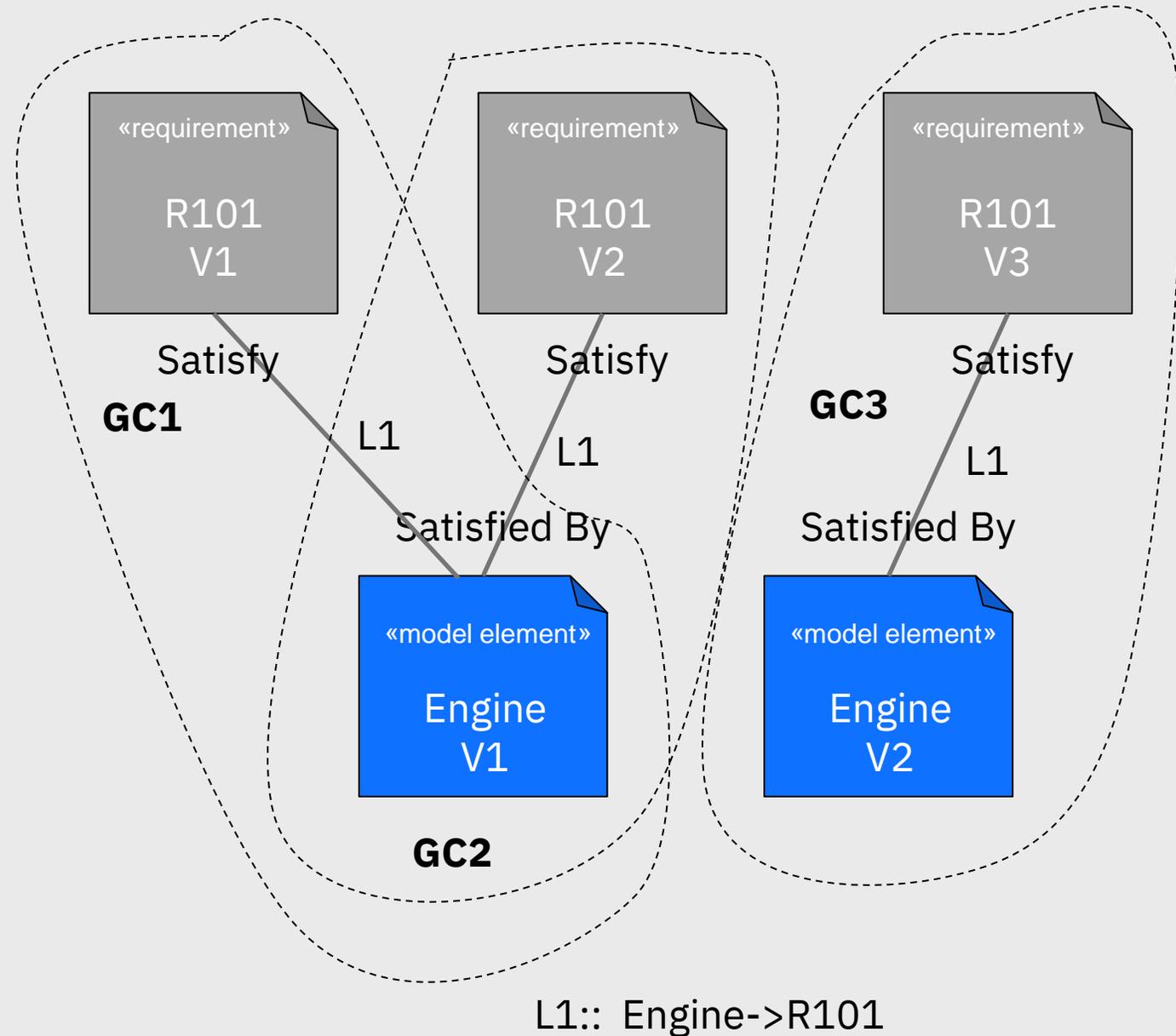
Bi-directional linking: link storage and link discovery

- Conceptually all OSLC links are bi-directional
 - Enable navigation from applications on both ends of the link
- Enabling bi-directionality:
 - Storing links at both ends: **unrecommended** practice
 - May easily lead to data-model inconsistency
 - Storing a single link with **“incoming link discovery”**
 - Avoids inconsistencies due replication of data
- Single link storage requires conventions on where link is stored and how the non storing application discovers the link
- Linking profile standardizes the protocol related to single link storage and incoming link discovery



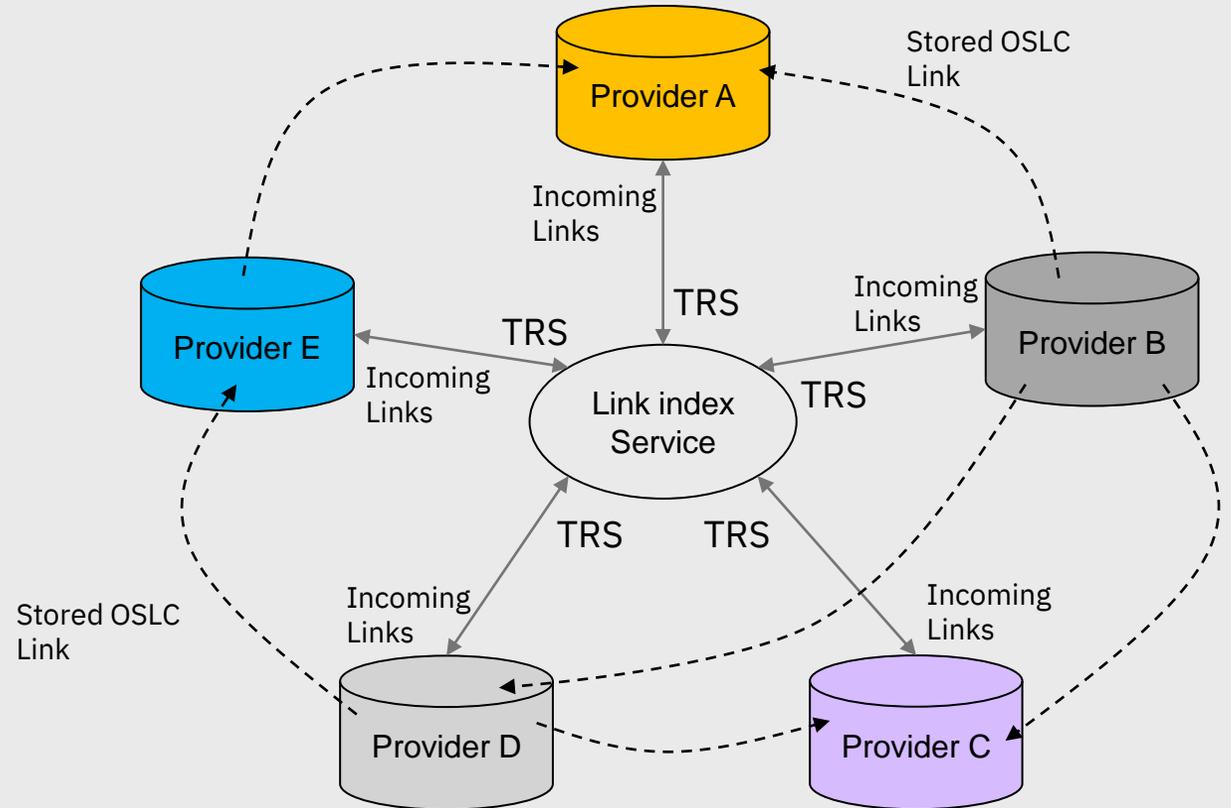
Linking in context of configuration management

- Linking is fundamentally impacted by global configuration management
 - Selecting target resources
 - Resolving target navigation
 - Discovering incoming links
- GC aware linking specifies mandatory requirements and provides guidance for linking in context of configuration management



Optimizing linking with link discovery service

- Inquiring all providers for incoming links does not scale well
- Link index service provides incoming links data in a single query
- OSLC Providers contribute link data to the service via TRS feed
- Currently discovering and querying the service is not standardized
- Requires a new OSLC spec and an extended linking profiles



Future related work

- Standardizing Link Index Service
 - Scalable way to explore incoming links for large OSLC provider network
- Standardizing Link validity service (TBD)
 - Maintain and inquire link validity (suspect link) status of OSLC links
- Completion of GCM spec
 - Standardizing changeset delivery for modifying data in OSLC GCM providers

Questions?