



Challenges to OSLC Link Management in Safety Audited Industries

Conference Impulse

- public -

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) for CRYSTAL – Critical System Engineering Acceleration Joint Undertaking under grant agreement n° 332830 and from specific national programs and / or funding authorities (BMBF 01IS13001E).

Why do we use links?

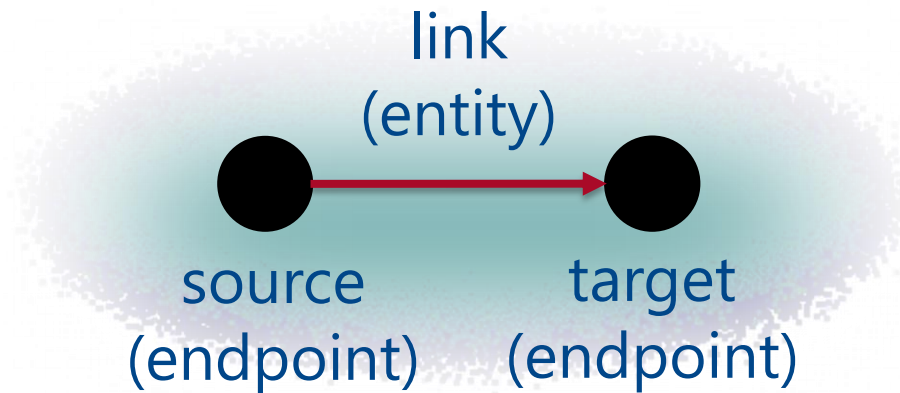
- Applicable standards
IEC 61508, ISO 26262, ISO 14971, DO-178B/C
- demand sound development process (quality + reliability)
- and credible proof of the process
- and credible proof of dealing with all safety concerns

- **Main problem: homo sapiens**
 - Humans perform complex transformations in development processes and those are neither reliable nor easy to document
 - Resulting items often stand in no easy relationship to each other

- One proof contribution: document the development activity (input, function, output) → this is the „link“ or „trace“

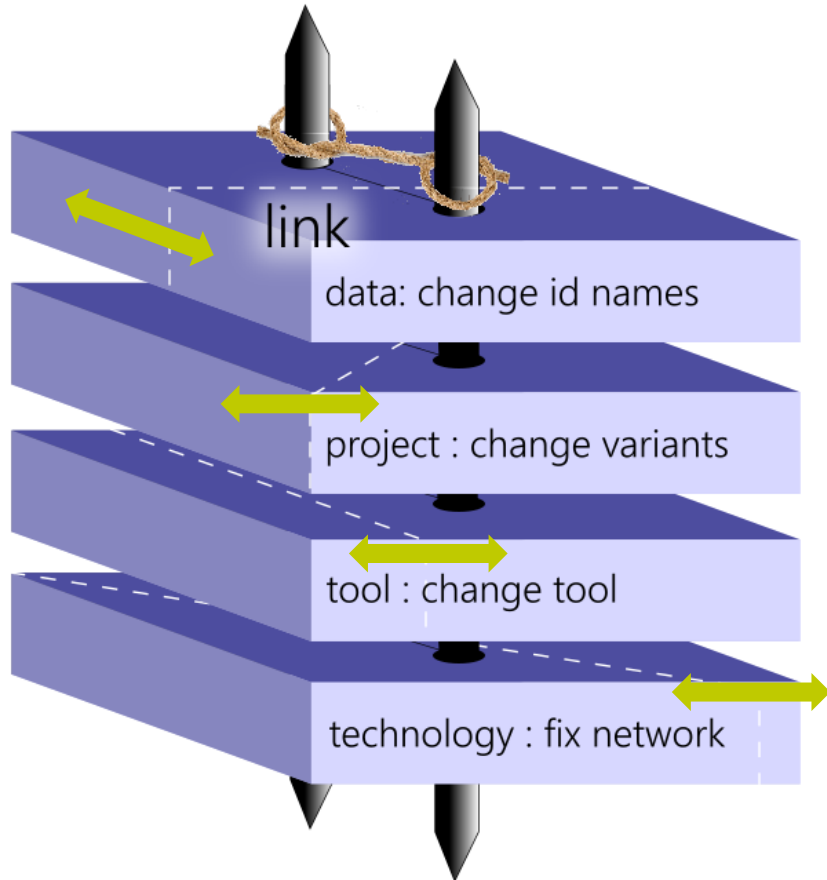
What is the Link?

- Links express an (possibly directed) explicit relationship

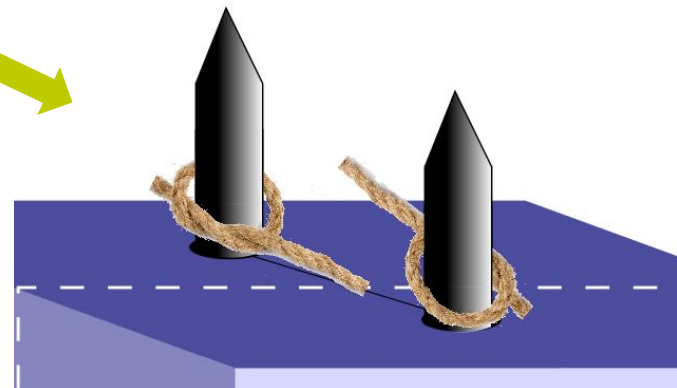


- You know from what to what the relationship exists
 - You can separate it from other relationships having the same source and target or just opposite direction
- ➔ one endpoint alone is not a link (and two not necessarily)

Links must withstand forces of change



- Links are affected by various layers of toolchain
- Imagine activities as „shearing forces“ on „rope“
- In some cases links must break by guarantee!



Links must withstand forces of change

Data

- Transport between tools
- Reorganization of data
- Synchronization of data
- Concurrent editing

Project

- Mesh-up of baselines
- New variants
- Reuse
- Changing traceability demand

Tool

- Changes of technical representation
- Different attributes
- Data distribution concept
- Semantic Interpretation

Technology

- Changes to network topology
- Changes to number and role of servers
- Developing Security
- Alternative Transport Layers

- Very often people ask about linking:
 - How do I add a link?
 - How do I navigate over a link?

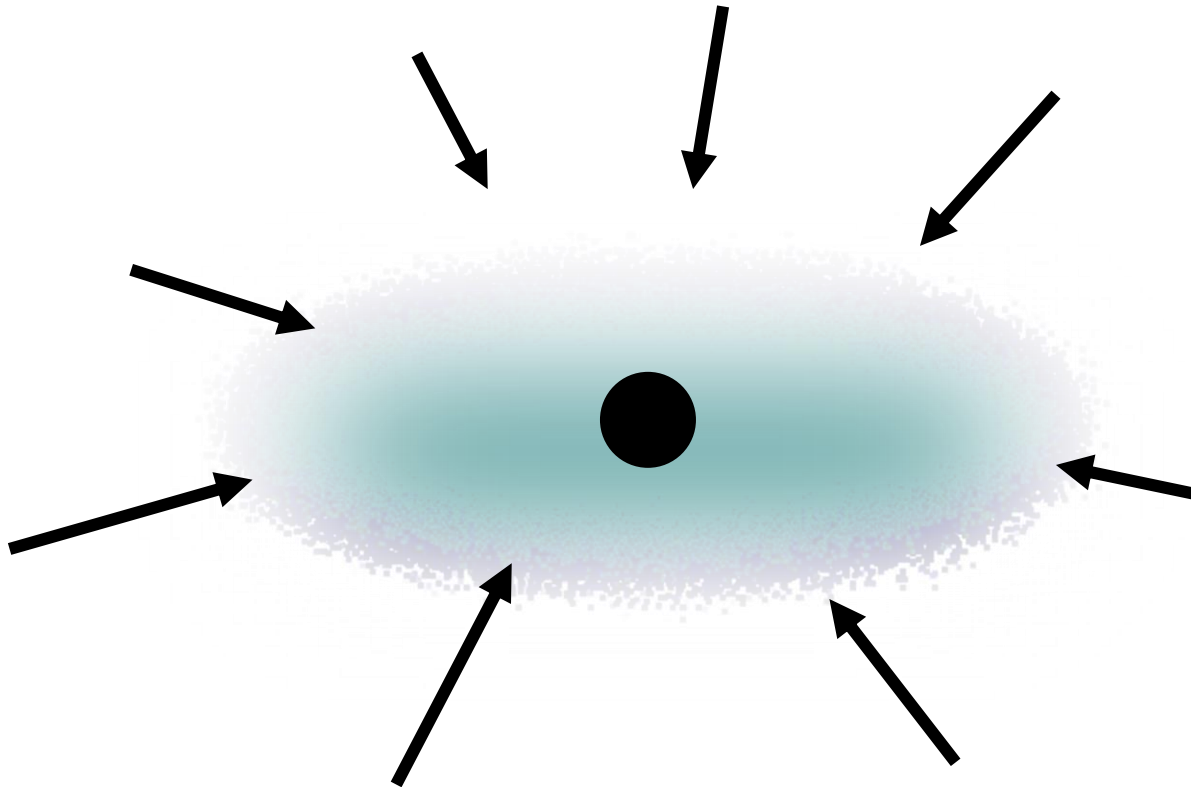
- But equally important questions are:
 - How do I remove links?
 - How do I redirect links?
 - How do I recognize link integrity?
 - How can I point to a placeholder?
 - How can I reproduce complex link structures?
 - Access rights to links?
 - How do I implement various instances of a link?
 - How do I make sure that links of certain types will only point to resources of certain types?
 - How do I make sure that links will not navigate across configurations?
 - How do I make sure that links will not navigate across baselines?
 - ...

What is the OSLC Link?

- OSLC involves two kinds of „links“:
 - Universal Resource IDs (URIs)
 - RDF links (in data packets)
 } None of it is the same as an „ALM link“
- OSLC intends to provide a „weak“ form of link over the boundary of ALM/PLM supervision
- OSLC claims only to provide „lose coupling“
- The „lose“ quality is crucial in deciding whether your projects must implement certain links on a firm platform or may implement them as OSLC links
- Note: you can have „link redundancy“
 - Firm links for auditing
 - Lose Links for daily work
 } Implies some automatic housekeeping
- With OSLC you can often link with great precision to items

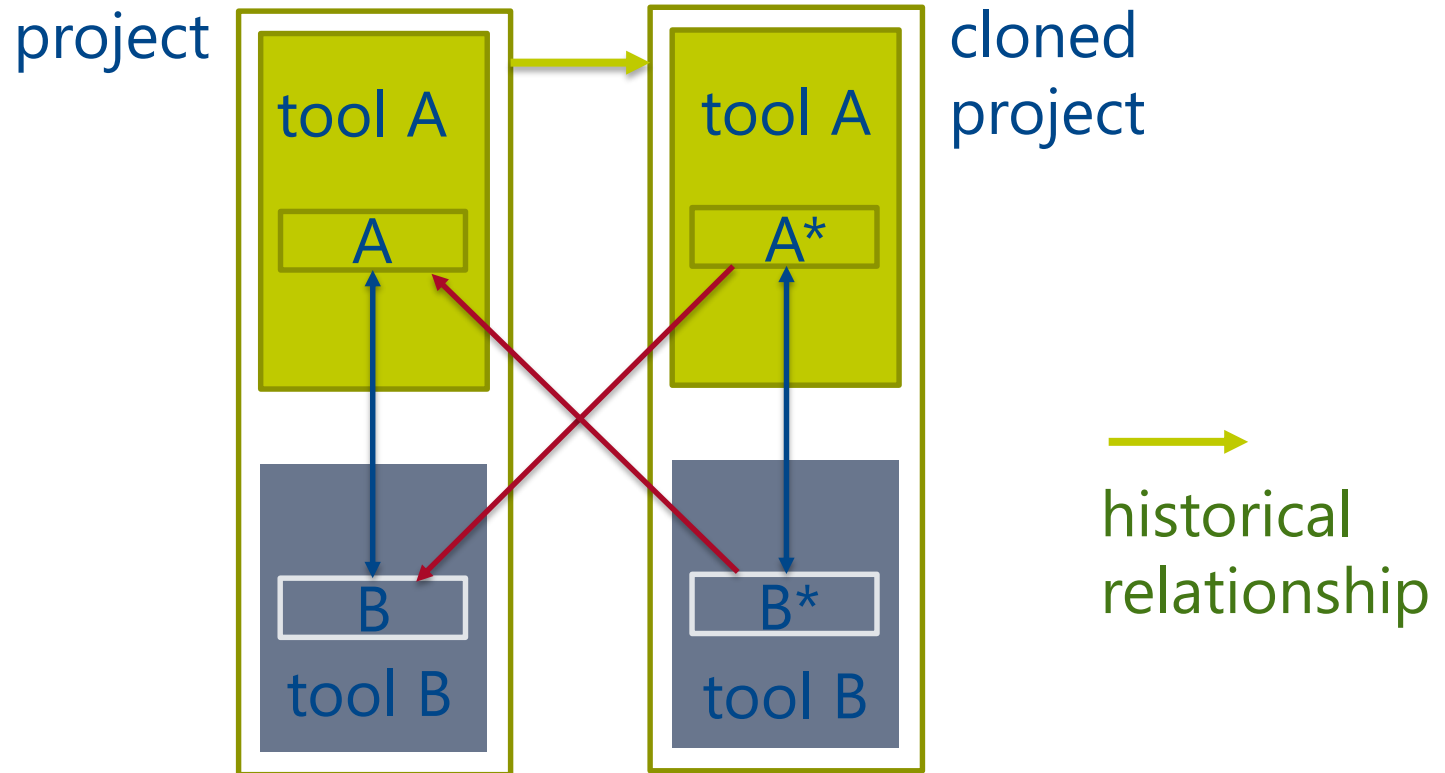
Hello? Anybody out there?

- Knowing a URI does not mean you can know what is related to it and why? → Where to query them all?



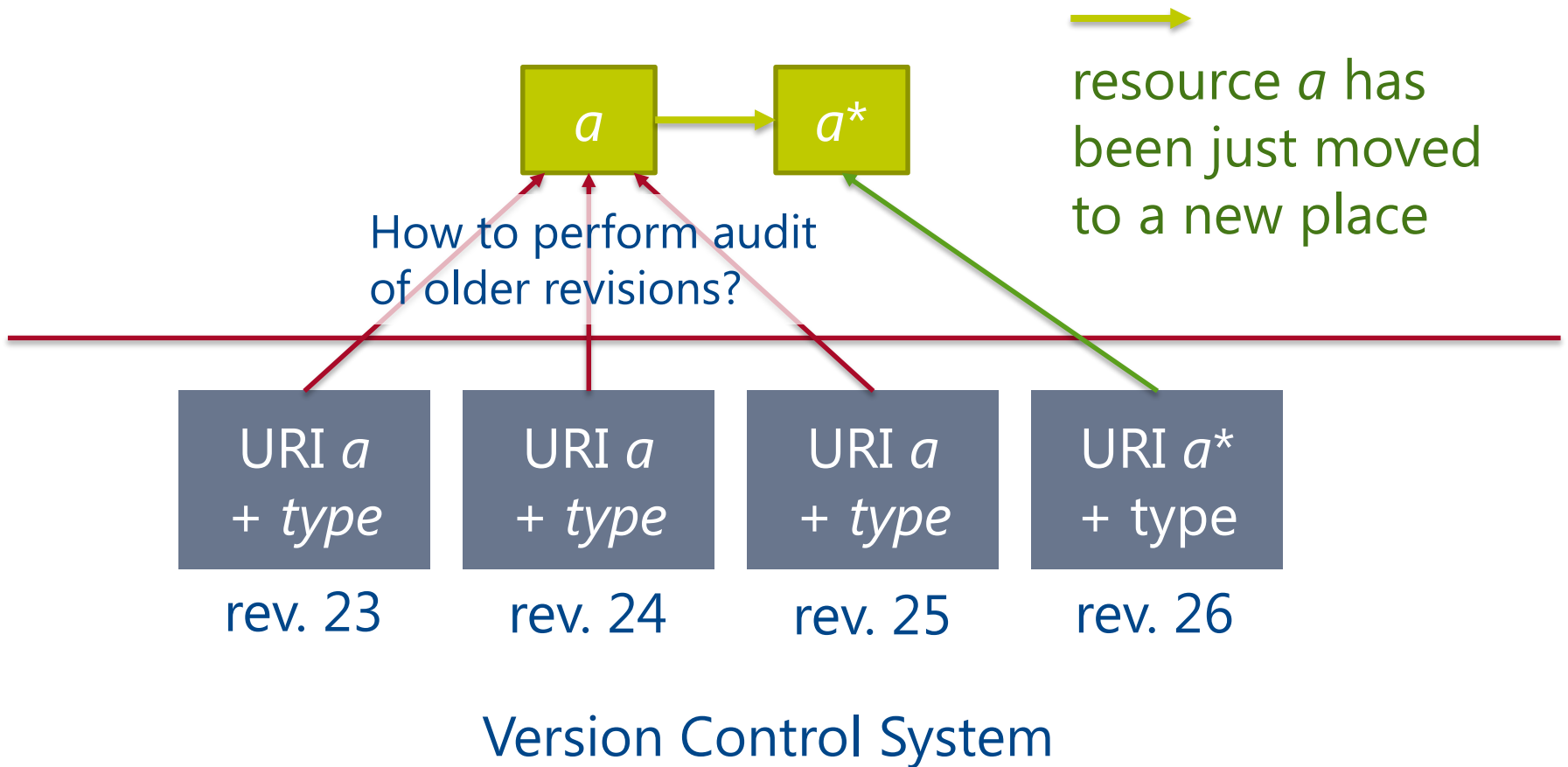
Absolute Linking

- OSLC URIs are „absolute“ by very nature
- Combination of techniques required in order to live with it



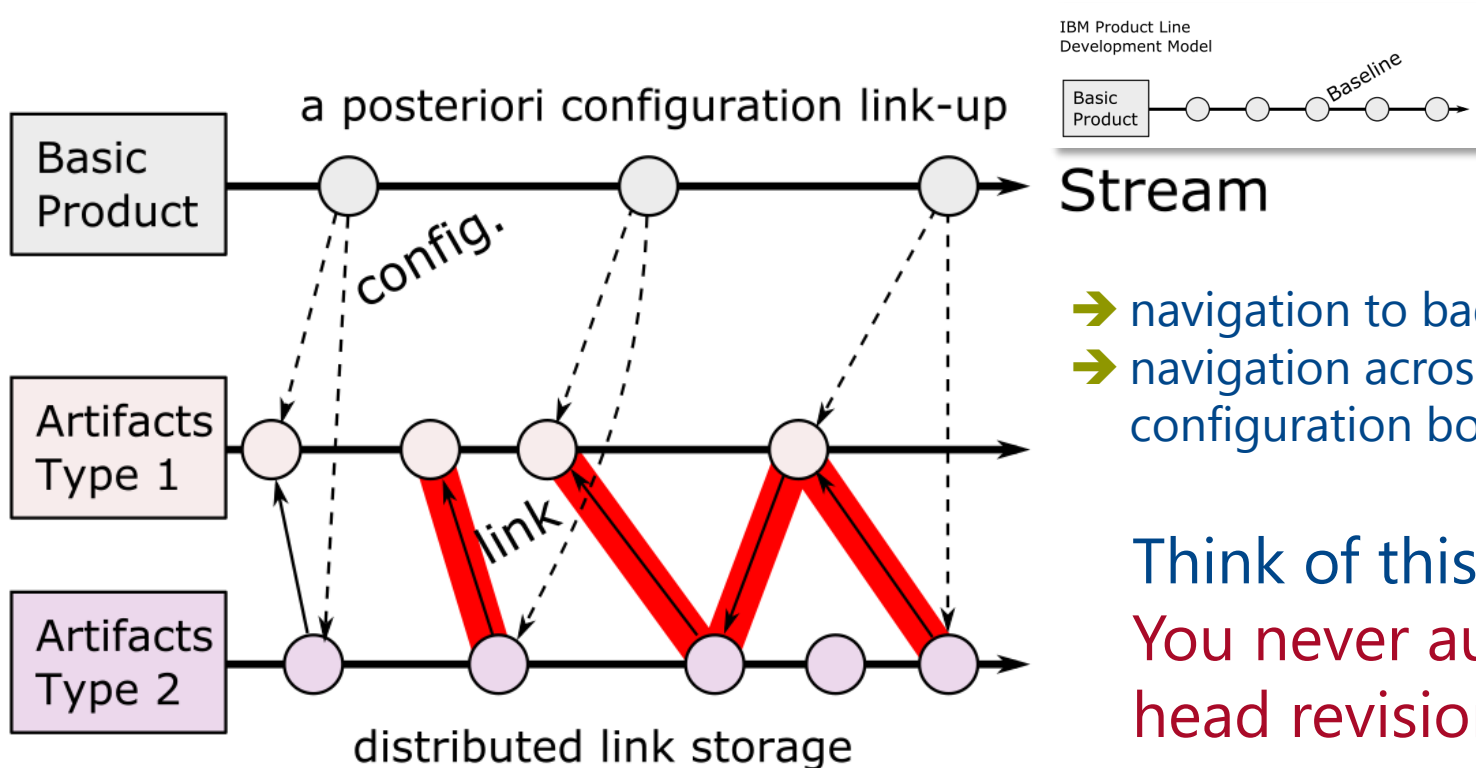
Storage matters

- URIs can be put in places where it is difficult to manage them



Survive the Reviews / Audits

- Without contextual link resolution (*get me the right link in this configuration, variant, baseline*) you will observe various link integrity problems with *distributed link storage*



- navigation to bad versions
- navigation across configuration boundaries

Think of this:
You never audit head revisions!

- OSLC is using technology of the web in order to
 - document relationships between artifacts.
 - quickly navigate to those items using universally available browsers.
 - uses most generic URIs as link endpoints.

- For Critical System Projects this means:
 - Linking within ALM (or comparable products) should be preferred
 - OSLC links are not naturally suitable for long term projects / products
 - Experience is showing that OSLC links do not withstand various types of „stress“ → careful application of them is mandatory

- Outlook
 - We will see more features in OSLC4J and OSLC specifications in order to simplify some link management in the future (e.g. OSLC CCM 3.0)



Thank you for your
attention!

