Swiss Re

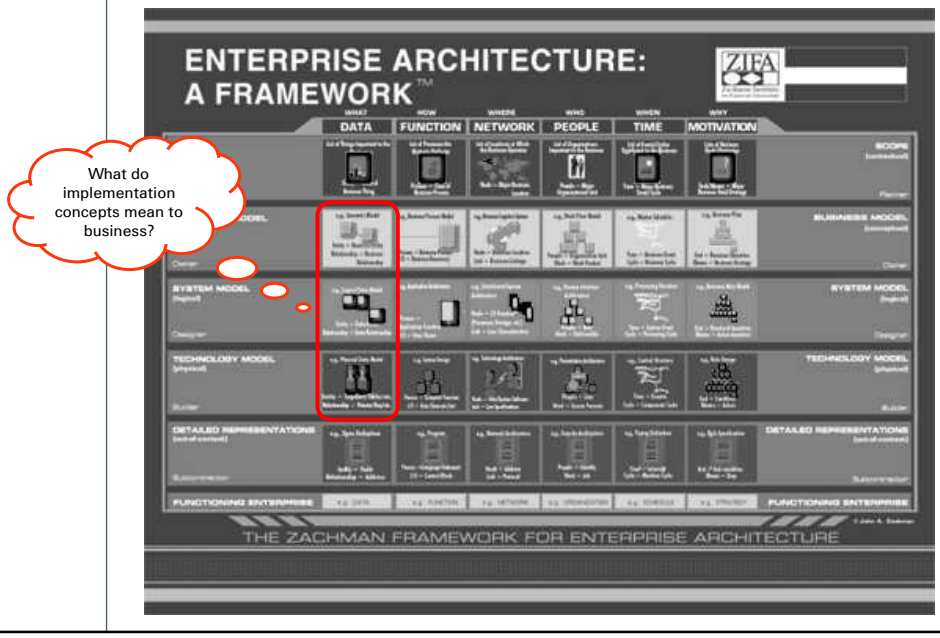
Multimodel Application Specification, Integration, and Evolution

Swiss Re

Problem and solution scope

Model integration across different levels of abstraction

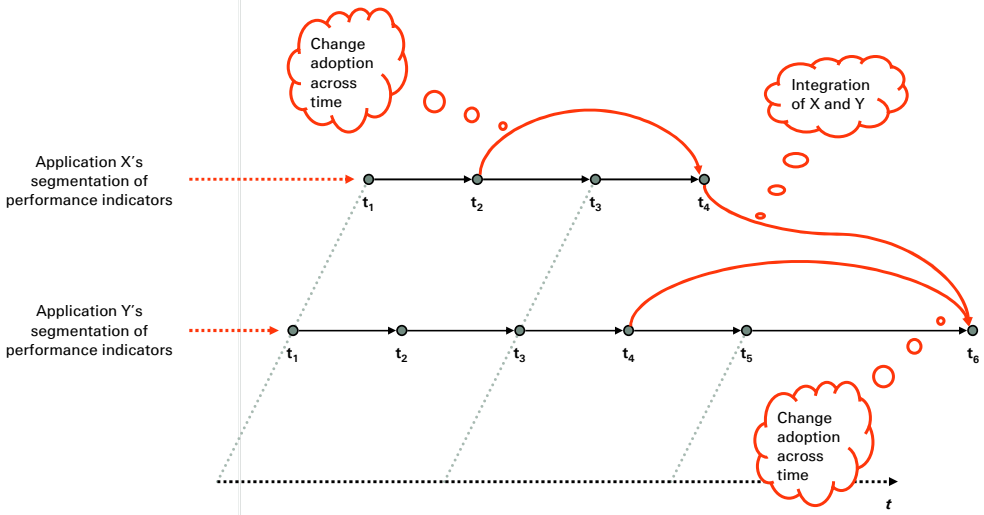
Swiss Re



Page 3

Model integration with variants and across time

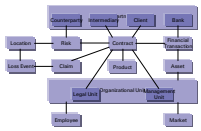
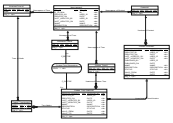
Swiss Re



Page 4

Swiss Re

Swiss Re Data Language

	Entities	Attributes
Business		■ Business terminology ■ Dimensional structures “to be” ■ Other reference data “to be”
System		■ Codes ■ Dimensional structures “as is” ■ Other reference data“as is”
Technology		■ INTEGER ■ SMALLINT ■ VARCHAR(256)

Swiss Re

Solution elements

Specification, Integration, and Evolution Framework

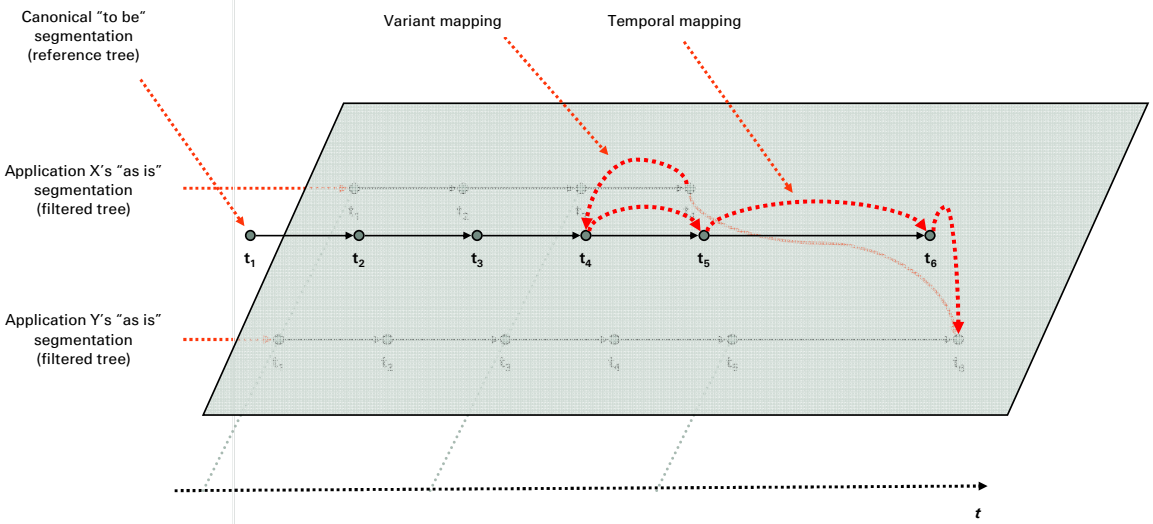
Swiss Re

- Distinguish model types and manage differently
- "To be"
  - canonical concept, business level
  - standardize as part of information architecture
  - govern implementers and manage evolution
- "As is"
  - implementation of applications, system level
  - relates to "to be" model
  - structure at discretion of implementer

Page 7

The canonical model as an arbiter

Swiss Re



Page 8

Separation of concerns in the information architecture

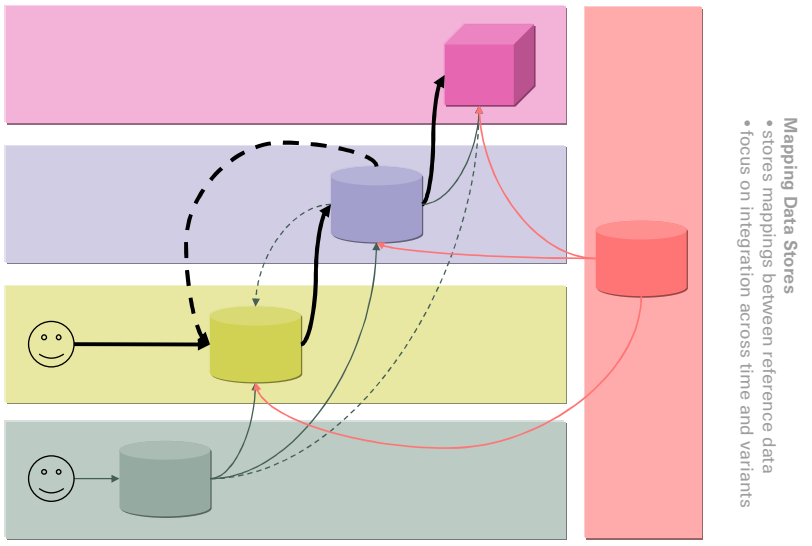
Swiss Re

Analytical Data Stores
- business decision-making
- data typically not forwarded
- focus on business steering processes

Integration Data Stores
- compile and exchange data
- integrates transactional data stores
- presents consistent view of business

Transactional Data Stores
- runs the day-to-day business
- focus operational business processes

Reference Data Stores
- store reference data
- focus on time and history

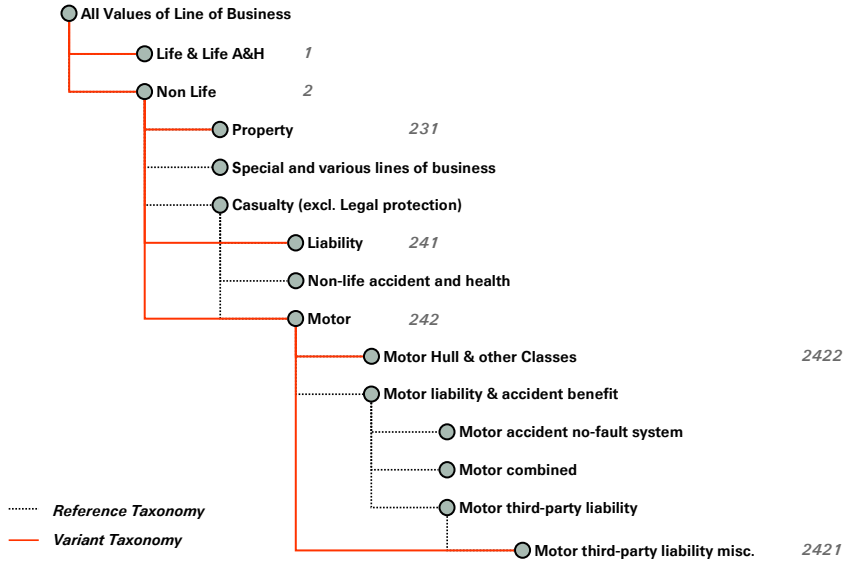


Modeling experiences

Swiss Re

Business vs. system model normalization

Swiss Re



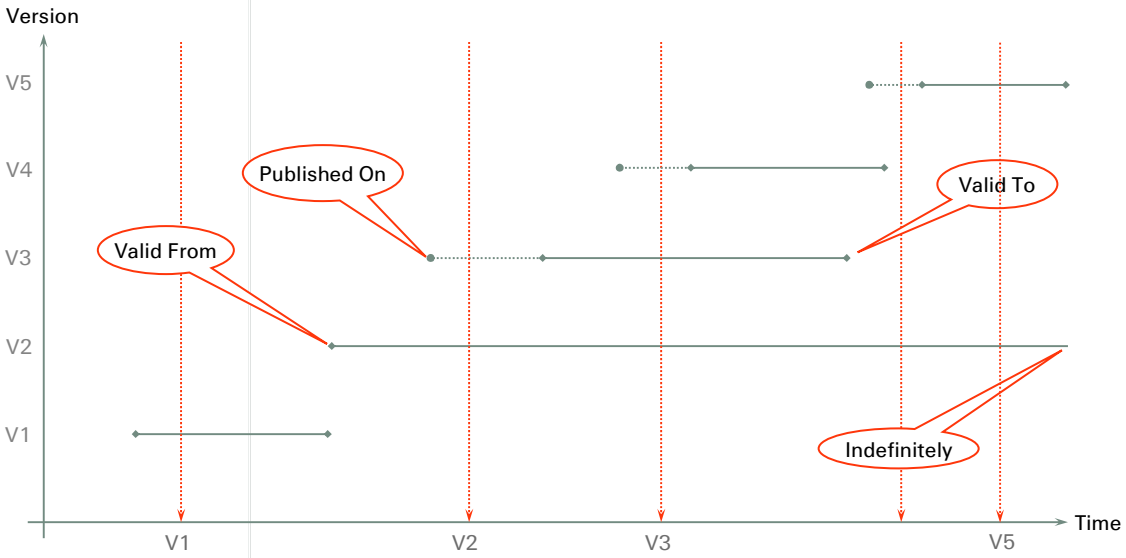
Simplicity vs. expressiveness

Swiss Re

- How to handle conflicting constraints from different domains
  - add business concepts to modeling language
  - allow expression as system concept
  - do not express as concept anywhere
- Approach
  - separate hard and soft constraints
  - exploit tacit (ambient) knowledge
  - handle inconsistency as part of architecture

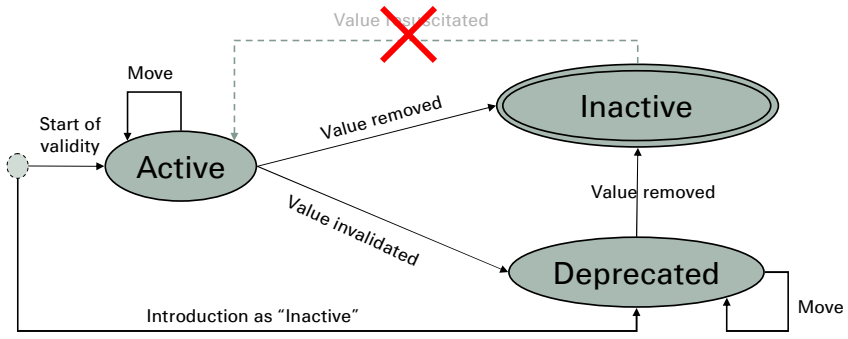
Versions and time: what version is valid when?

Swiss Re



Repository object lifecycle

Swiss Re



Repository object lifecycle and mappings

Swiss Re

Status	Business: Definition/Intent	SDL Tool: Interpretation
Active	Value is in use	Valid value in valid dimension(*)
Inactive	Value may no longer be used; must be mapped if mapping is available, otherwise must be rejected	Value no longer in dimension
Deprecated	Value is transitioning to not being used any more; may be mapped if mapping is available, but does not have to	Invalid value within valid dimension

(*) dimension = reference tree or filtered tree

Current work and outlook

Swiss Re

Swiss Re

Current challenges and outlook

- (De)Normalization and temporal mappings
- Identification of business-relevant evolution steps
- Jurisdiction

Swiss Re

Thank You! Questions?

Swiss Re

Dimension lifecycle events