Developing Architecture in Volatile Environments -

Lessons Learned from a Biobank IT Infrastructure Project

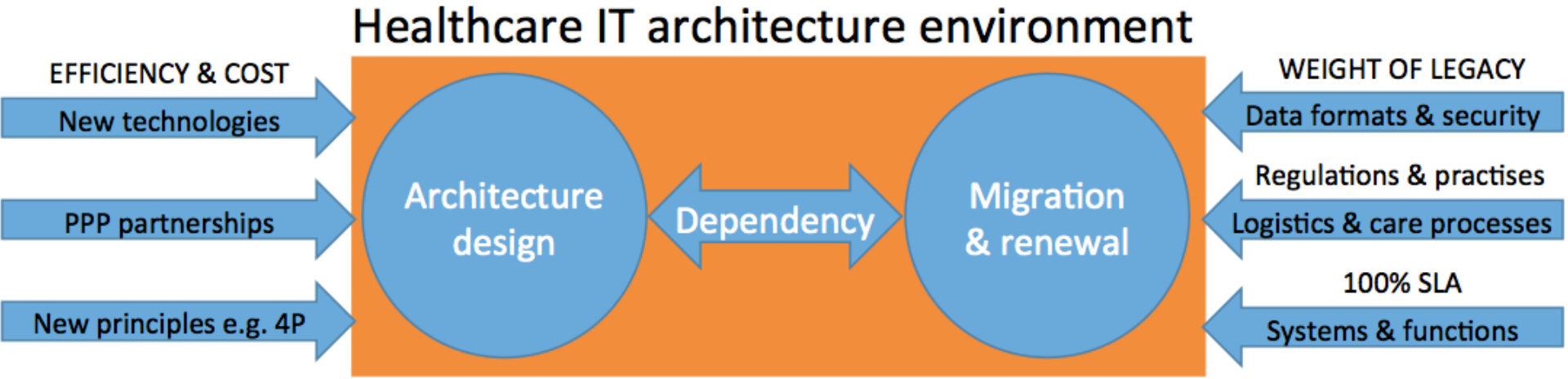
Hyysalo, Harper, Sauvola, Keskinarkaus, Juuso, Salminen & Partala

Faculty of Information Technology and Electrical Engineering

University of Oulu

Introduction

- In volatile environments an iterative architecture process is required, where technical and non-technical concerns are separated.



- An architecture process that is modular, interoperable, controlled and abstracted is suggested.

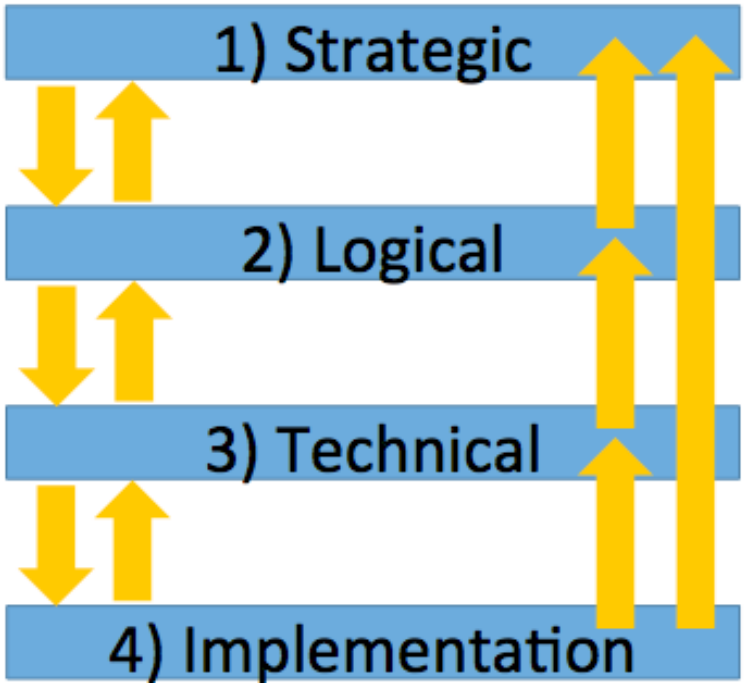
Research question

- There is a need for an architecture process that addresses:
  - evolving and volatile environments
  - wide scope of the project and several different stakeholders
  - new models and regulations
  - continuous renewability
- Research question:
  - ***What form of architectural design process is suitable for volatile environments?***
- Collaborative effort with:
  - Finnish biobanks, software and system providers, standardisation organisations, hospitals, …

Four architectural levels

Continuous Renewability approach – 4 levels:

- Each level presents a view to a development
- Each level implements the level above
- Each level is a design phase
- Each level corresponds a set of stakeholders



1) Strategic architecture

- Starting point
  - Overall description of the development problem
  - Business views
  - Business processes and rules
  - Performance goals
- Defines the conceptual architecture with:
  - Visions
  - Organisational strategies
  - Business drivers and goals
  - Processes
  - Functional perspectives
- Provides the rationale
- Output: Semantic model

2) Logical architecture

- Defines the functions, resources and components including:
  - Relations and information flows
- Defines the qualities and measurements for achieving business goals
- Input:
  - Strategic architecture
  - Available building blocks
- Output: Logical data model

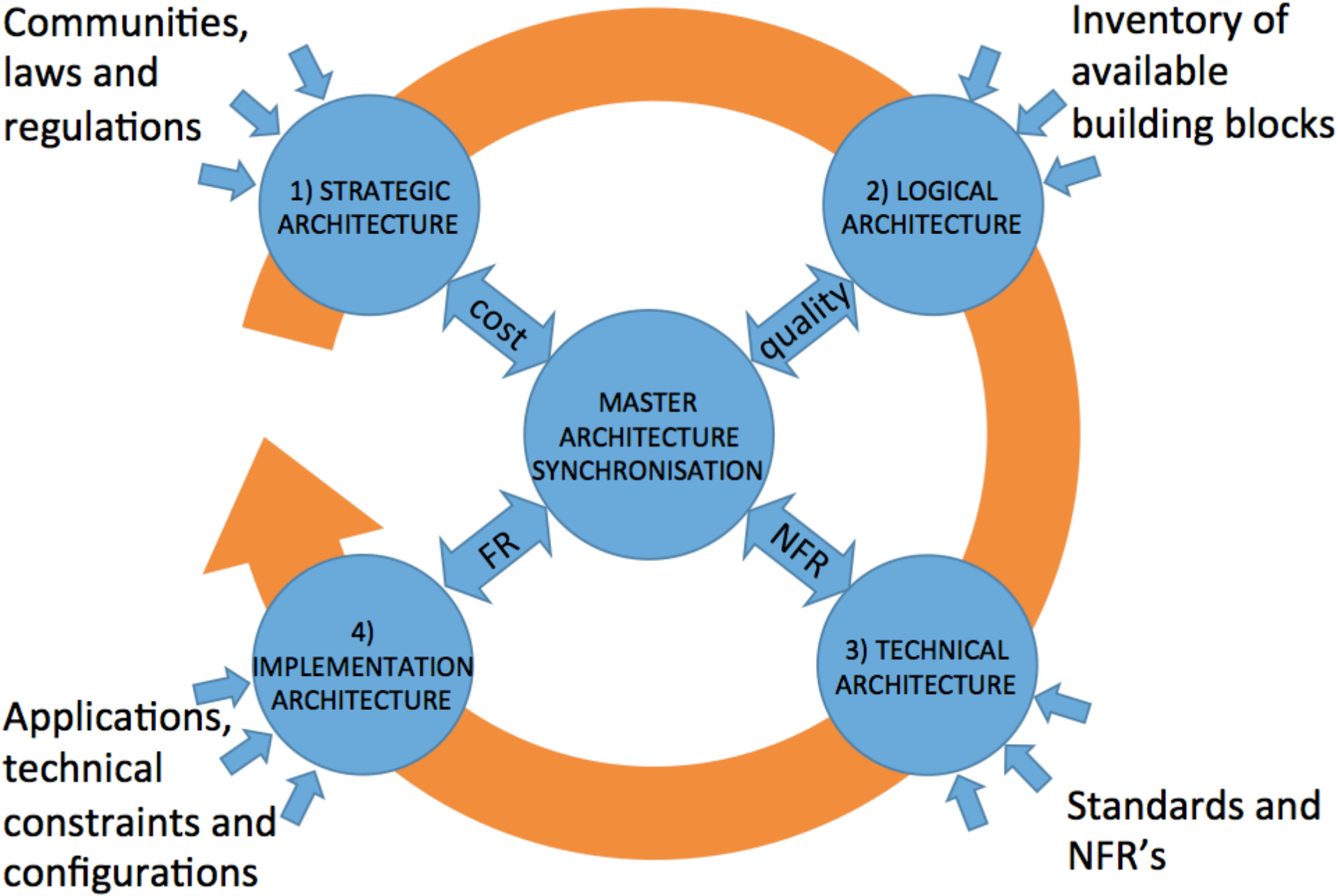
3) Technical architecture

- Implements the logical architecture with:
  - Technology platforms
  - Information system environments
  - HW & SW
  - Interfaces
  - Etc.
- Input:
  - Standards
  - Non-functional requirements
- Output: Definition of the physical data model and a technology architecture

4) Implementation architecture

- Implementation architecture focuses on:
  - HW & SW
  - Operating systems
  - Middleware
  - Interoperability
- Input:
  - Configurations
  - Technical constraints
  - Application requirements
- Output: Definition of the implementation details, components, applications, HW & SW configurations

Continuous Renewability architecture



Evaluation of the architecture

- Generalisable
- Utilise open-standards and protocols
- Separation of concerns
  - Separation of non-technological issues from technological issues
- Adabtable
- Limits complexity and the effects of changes
- Scope, guidance and verification checkpoints
- Modular

Key benefits

- Easy adaptation to future needs by localising changes to only the affected domains.
- Ability to choose any software to specialise each domain
- Not relying on one vendor.
- Enabling open source solutions reduces maintenance and licensing costs.
- Modularity allows for better innovation opportunities.

Thank You!

Questions?

jarkko.hyysalo@oulu.fi
gavin.harper@student .oulu.fi