SOA Governance

Experience Sharing on the eHR Project

Clara Cheung
Pegasus But
Warren Cheung

30 Apr 2012

Agenda

- SOA in the Hospital Authority
- Design Time Governance
- Run Time Governance
- Some recommendations

Service Oriented Architecture in HA

Service Oriented Architecture in HA

- Adopt SOA since 2006
- Establish Enterprise Architecture Platform on UNIX
- Build ‘technical services’ and ‘business services’ for the Clinical Management Systems (CMS)
- Fully adopt SOA in CMS III

electronic Health Record (eHR)

- Hong Kong Government Project
- HA served as the technical agent to build the HK wide eHR sharing among public and private healthcare providers
- Leverage on HA's systems and experience
- Adopt SOA on Linux

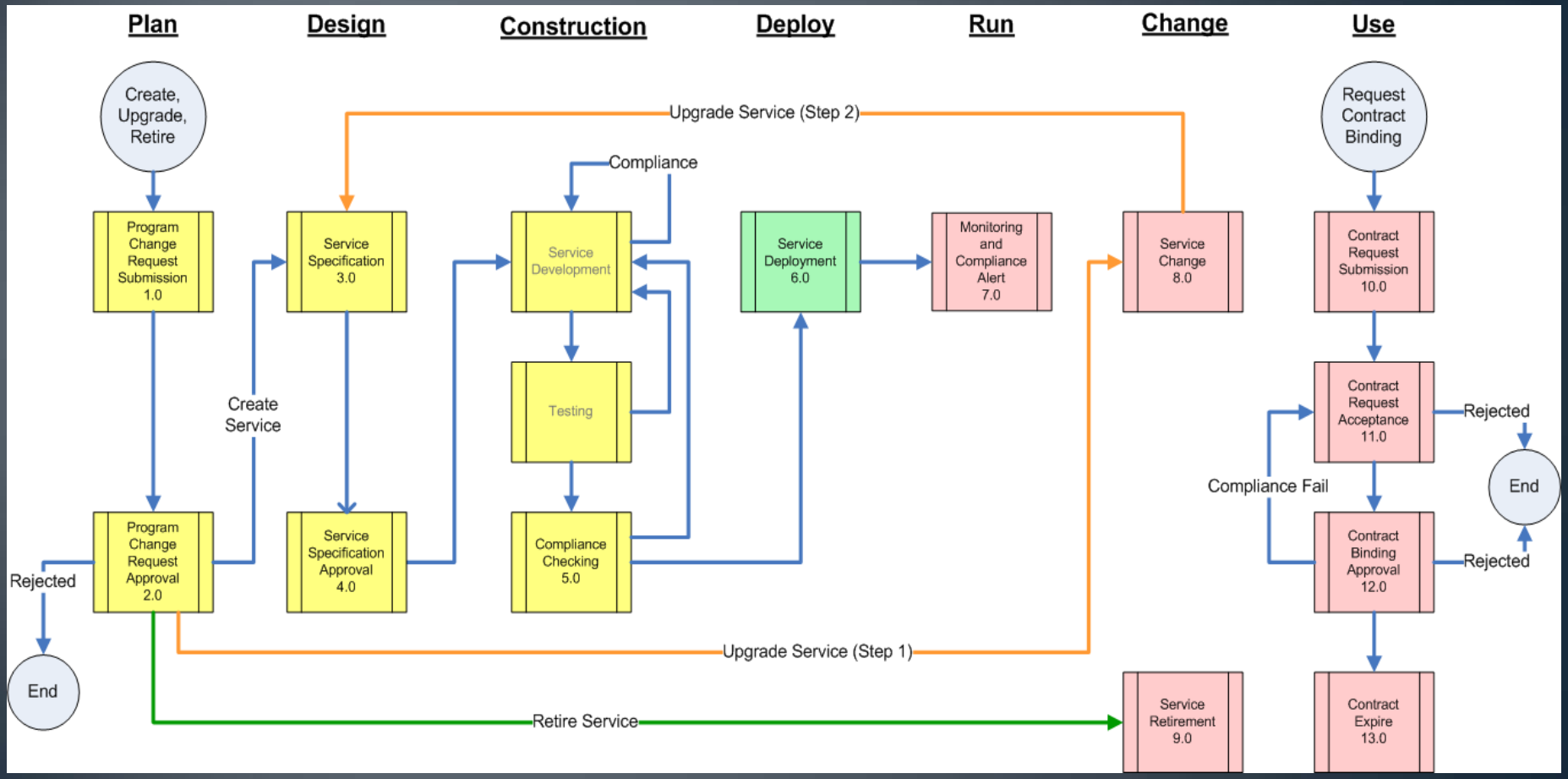
SOA Lessons learned from CMS

- Technically challenging
- 'Collaborative Intensive' which means very high communication overheads
- Significant overheads before benefits can be realised
- Often unclear roles and responsibilities for shared services
- Not easy to persuade developers to rely on others
- Easy to have proliferation of similar services
- May have well intentioned services developed but not used
- Performance affected by crossed sites or crossed servers navigation among services
- Control of service lifecycle is difficult
- Tends to lose track of Provider/Consumer relationships

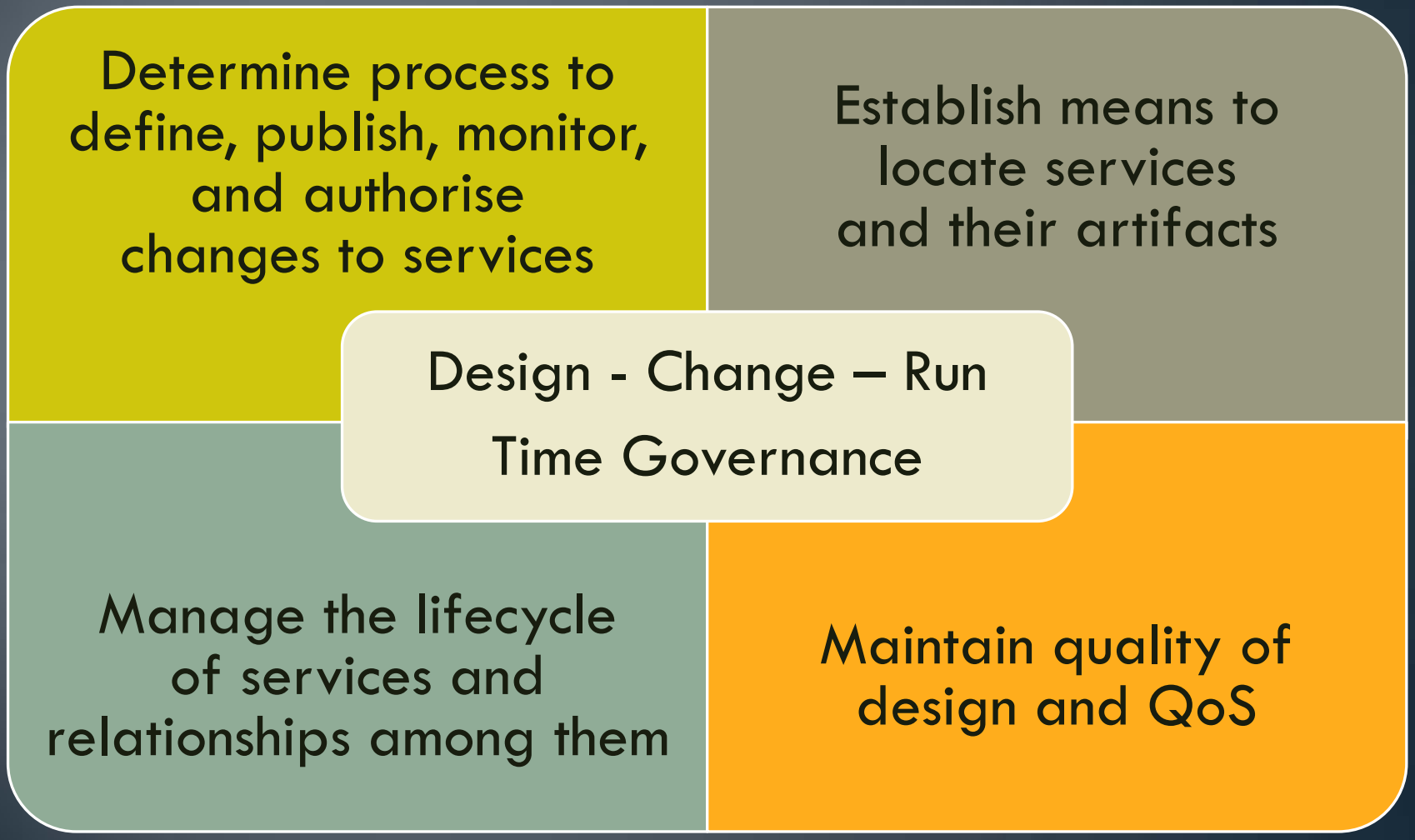
Lessons Learned

- Commitments from senior management to technical staff
- An architect in a senior capacity with authority is a must for SOA projects
  - To ensure consistency and integrity among modules / systems of the whole project
  - To be the umpire when there are uncertainties or compromises
  - To make the difficult decisions
  - Sometimes ‘top down’ is the only way

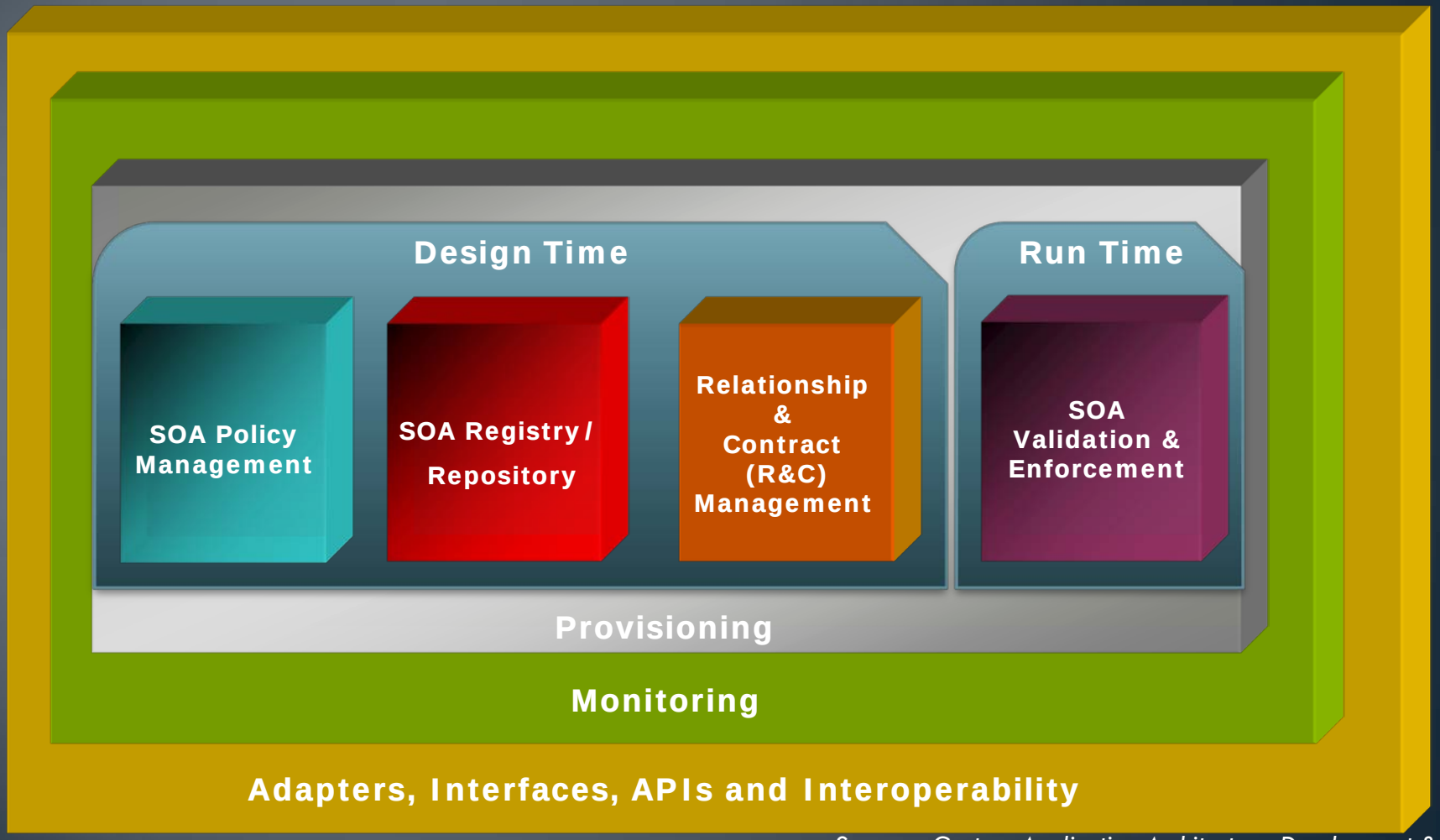
Governance Process Workflow



SOA Governance Objectives



SOAG - Principles



Source: Gartner Application Architecture, Development & Integration Summit, 2008

SOA Design Time Governance

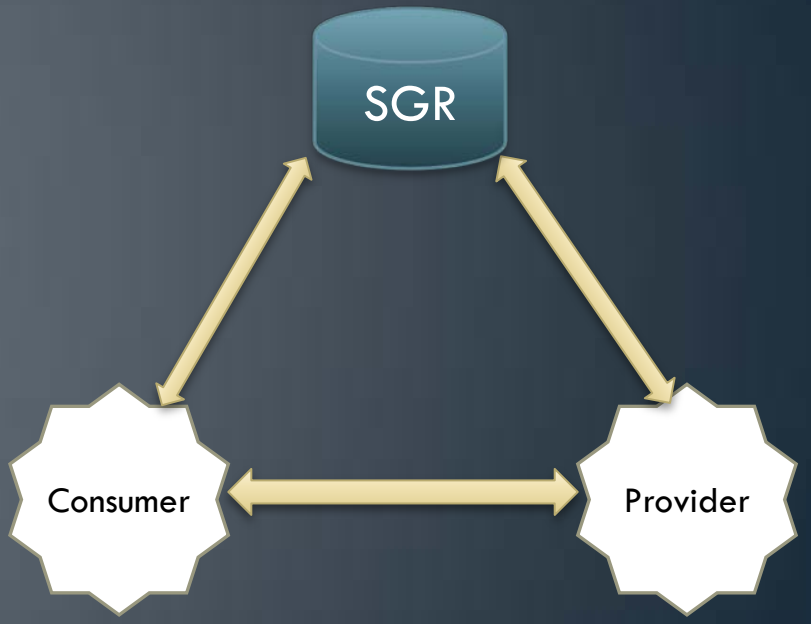
SOAG Workflow

SOA Policy Management

- SOA Governance structure needs to be defined as early as possible
- Governance to
  - Determine needs for a shared service
  - Who will be the provider
  - Who can be the consumer
  - Manage changes and retirement of services
- To what extent should business be involved?

SOA Registry and Repository

- Market research in 2010
  - Commercial products usually good at design time governance, but
  - Lack Adapter, API, library for Run-Time Governance enforcement
- In-house Developed : SGR (SOA Governance Repository)
  - Centralised Services Registry
  - Referenced by both Consumer & Provider Services
  - Integrate with existing SDLC and software migration workflow



SOA Governance Repository (SGR)

- A centralised service repository for the eHR project with objectives to :
  - Prevent service duplication
  - Optimise service granularity
  - Track relationship and versions between service providers and consumers
  - Allow service discoverability
  - Ensure objectives and requirements reflect SOA benefits, such as reuse and loose-coupling
  - Ensure proper documentation of service relationship
  - Ensure proper documentation & control of service lifecycle and state
  - Enforce service lifecycle management workflow

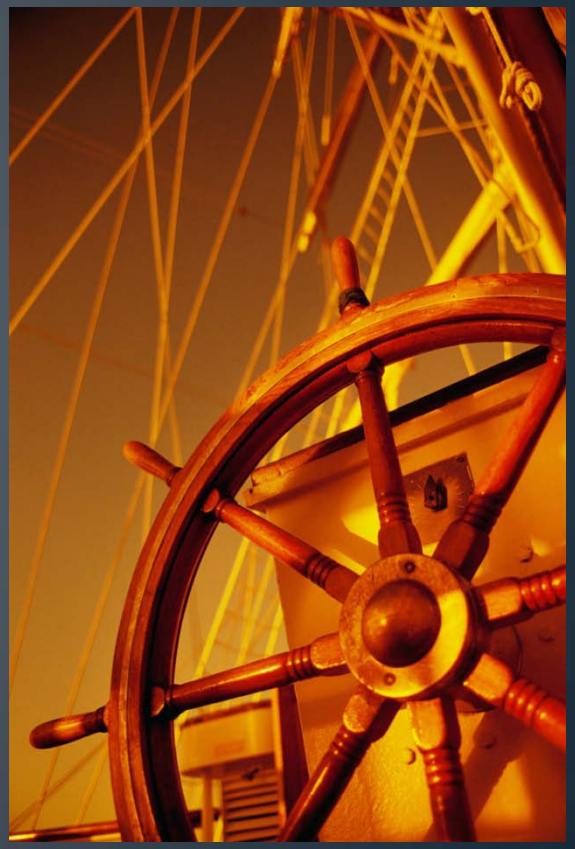
SGR

- Functionality
  - Provider Services Registration
  - Consumer Services Registration
  - Contract Registration
  - Lifecycle Management

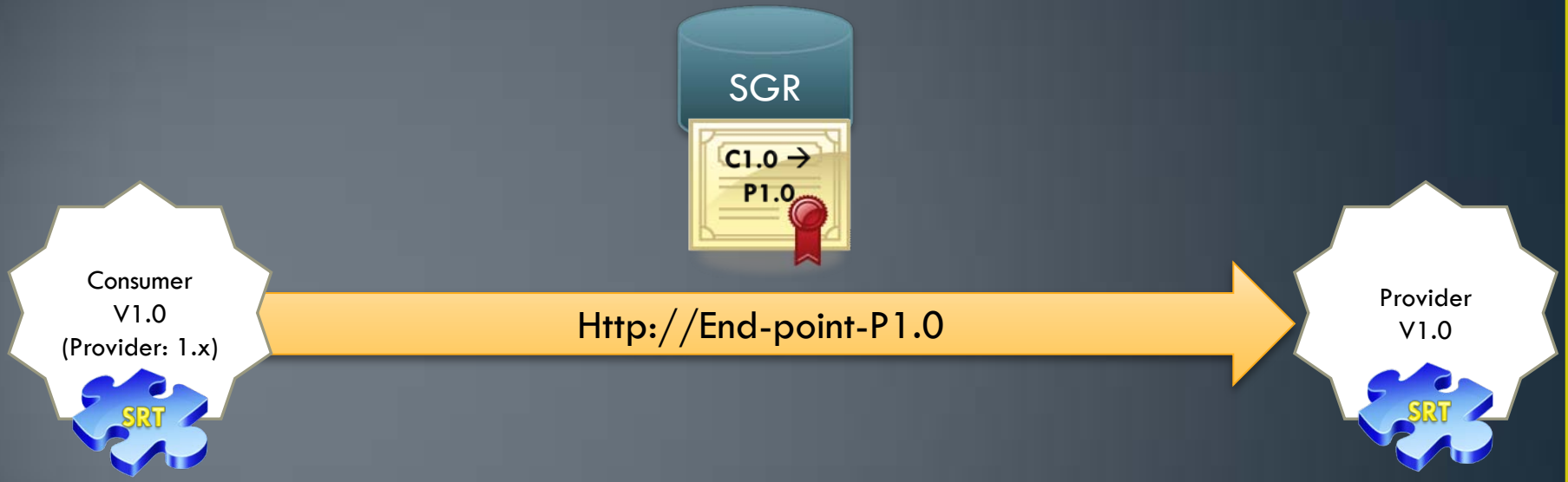
Run-time Governance

SOAG – Run Time Governance

- In-house developed library (SRT)
- Library based
- Contract based
- Supported protocols
  - WSDL
  - RMI/T3
  - Non-Java (like .Net, using Web-service Mode)
- Support high availability model of Application Servers in eHR
- Avoid program change on consumer side with minor service version upgrade on provider side (minor changes)



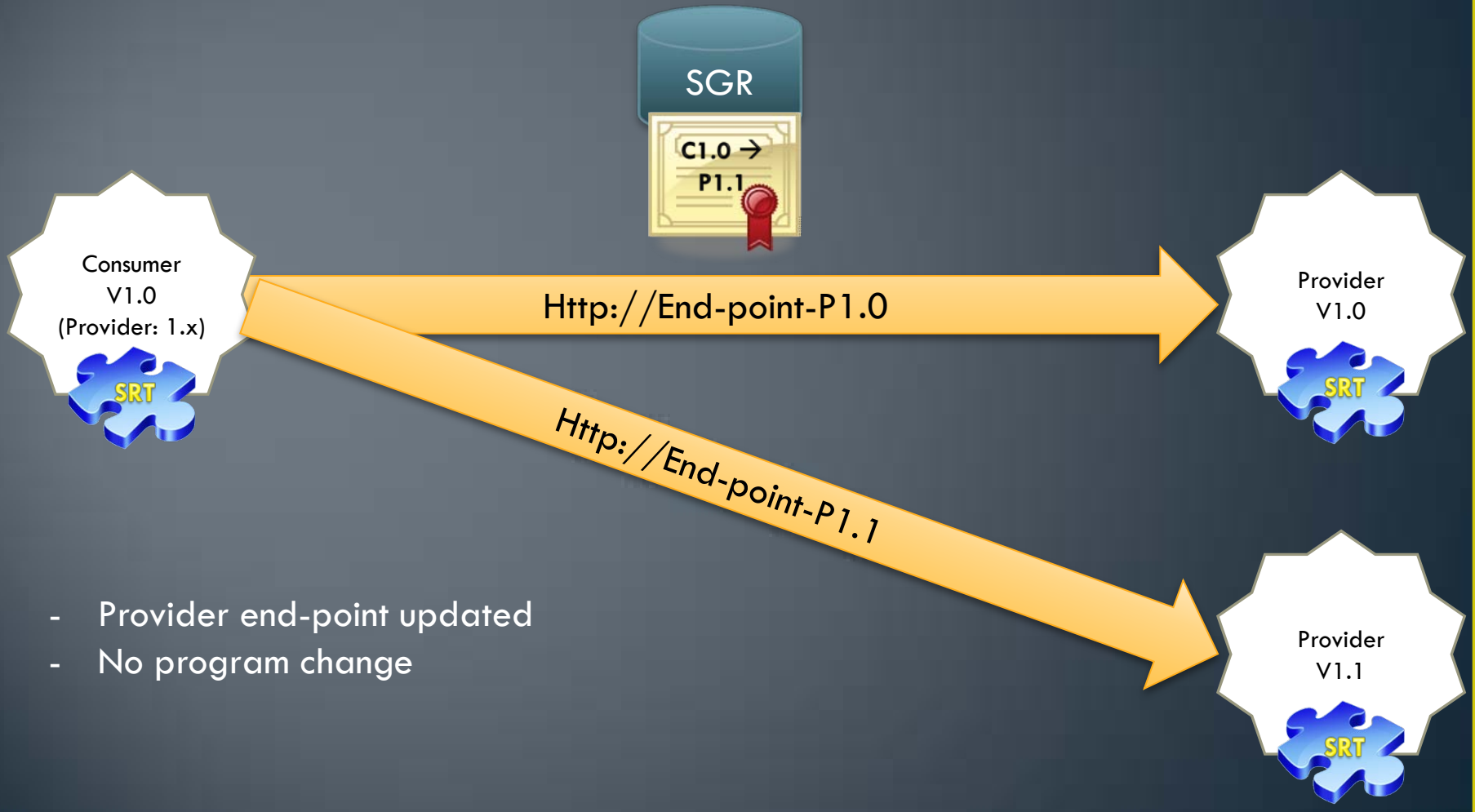
Workflow – Initial Stage



- Provider end-point

- Consumer authorisation
- Content filtering
  (WSDL only)

Workflow – Version Upgrade (Minor)



- Provider end-point updated
- No program change

SRT – Facilitate High Availability Model

- High Availability Model for eHR
  - Will have Primary and Secondary sites
  - Separate Domain for different sites – avoid traffic between sites
  - Local site domain cluster
- End-point lookup
  - Enforce priority lookup
    - 1st – Local host
    - 2nd – Local Cluster
    - 3rd – Remote Cluster

SOA – Hazard

- Service provider has to provide different but similar services with different elements
- Too many similar services deployed
  - Difficult to manage
  - Version control and service retirement become difficult

→ Content Filtering

Content Filtering - WSDL

PIP Available Elements	Consumer A Accessible Element	Consumer B Accessible Element
PATIENTS	✓	✓
- PATIENT	✓	✓
- EHR_NO	✓	✓
- NAME	✓	✓
- ENG_NAME	✓	✓
- LAST_NAME	✓	✓
- FIRST_NAME	✓	
- CHI_NAME	✓	
- LAST_NAME	✓	
- FIRST_NAME	✓	
- SEX	✓	✓
- PHONE	✓	
- DOB	✓	
- EMAIL	✓	

Some Recommendation

Before launching on an SOA project

- Is there a real need or will there be real benefits to adopt the SOA approach ?
  - Silo applications vs. group of inter-related / interactive applications
  - Are there any true value-added shared services?
- Are there senior management buy-ins?
- Are there technical staff buy-ins?
- Does technical leadership exist in-house?
- Will there be an umpire with adequate authority?
- What is the management expectation of SOA?

DOs

- Think Global, Act Local
  - Must have an Enterprise Architecture overview but may not need to wait for complete and detailed EA definition
  - Start on parts that are of manageable size and then define in details
- Identify your technical lead & umpire
- Consider a dedicated framework / shared services team
- Integrate your Service lifecycle management with your normal SDLC processes
- Before building a shared service, consider
  - Cost and benefit
  - Who is the provider?
  - Who is the consumer(s)?
  - Shared library vs. shared services
  - Avoid cascaded services
- Be flexible and common sense approach
  - do not only follow books
  - Be pragmatic about sales hypes