Increasing Trust in Public Service Delivery

Contract-Based Software Infrastructure for Electronic Government

United Nations University IIST

Center for Electronic Governance

Adegboyega Ojo
Tomasz Janowski

{ao,tj}@iist.unu.edu

Problem

1	Developing a TRUSTED Software Infrastructure for e-Government, where …
2	The Infrastructure comprises a MANAGED set of components and services supporting the development and execution of Electronic Public Services, and …
3	Management is carried out by specification, monitoring and mediation of Quality of Service (QoS) CONTRACTS between infrastructure elements and external client applications.

Outline

Public Service - Delivery

| Public service delivery is one of the three major functions of any government:	
1	Provision of common good including *public services* and infrastructure
2	Governance of the state, specifically rule making, implementation and adjudication
3	Maintenance of social order and security

Public Service - Types

| Four types of Public Services (Governance Enterprise Architecture – GEA):	
Certification	Issuance of documents to assert the state of an entity (citizen, business, etc.)
Control	Monitoring and mediation to ensure compliance with norms/directives
Authorization	Providing permissions and titles to the entities to carry out specific activities
Production	Provision of public utilities

Public Service - Concepts
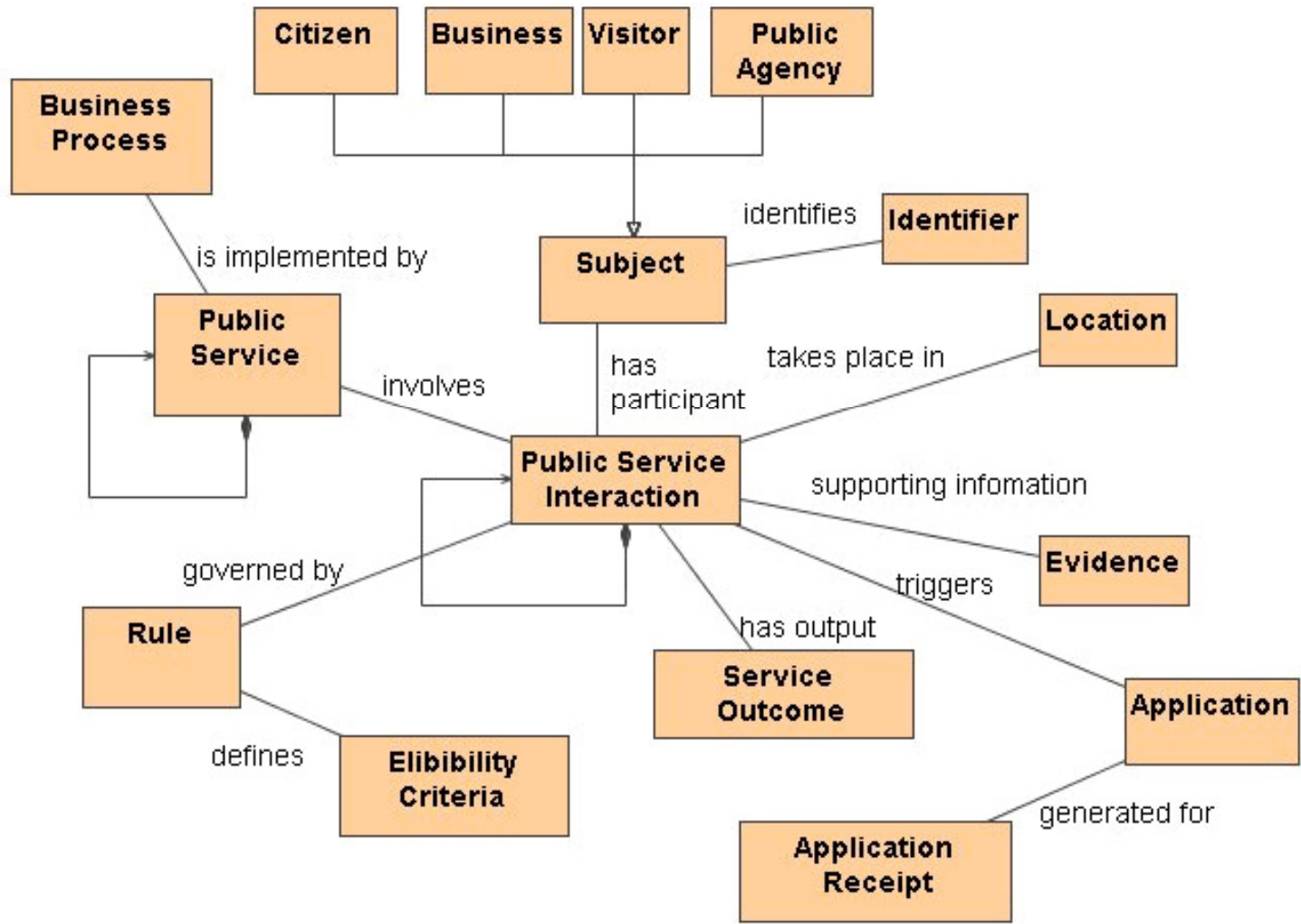
Public Service - Process

	Generic process for authorization-type of public services:
1	Application Submission
2	Application Validation
3	Eligibility Check
4	Internal Evaluation
5	Third Party Evaluation
6	Decision Making
7	Issuance/Denial
8	Notification
9	Archiving and Closure
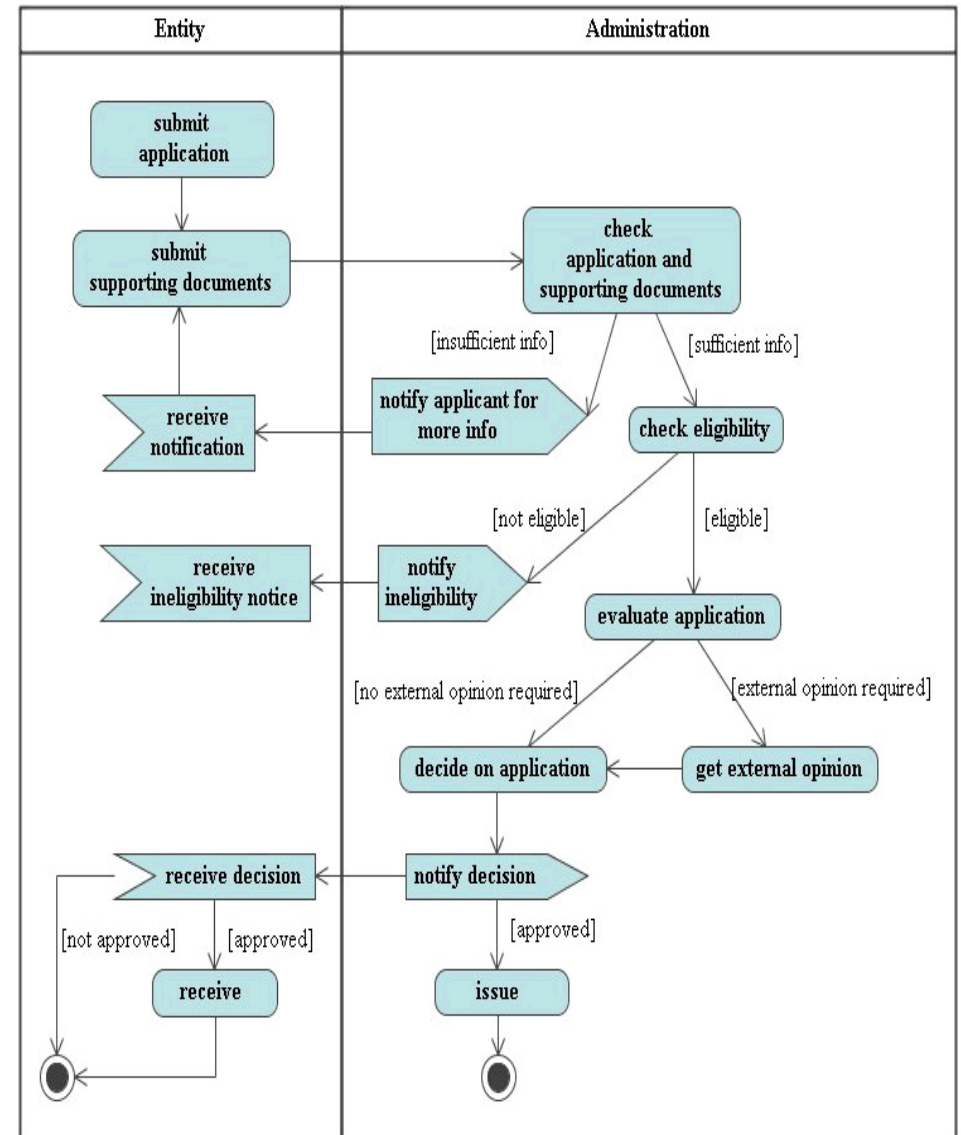
Public Service – G2C Examples

Welfare Services for Citizens	
1	Social houses
2	Financial aids to students
3	Retirement pensions
4	Financial assistance to individuals
5	Postgraduate scholarship
6	Survivor pension

Public Service – G2B Examples

Licensing Services for Businesses:

1	Temporary/Permanent electricity license	13	Food and beverage license
2	Construction and utilization license	14	Radio network license
3	Aviation industry license	15	Radio station license
4	Certificates of origin license	16	Nursing home establishment license
5	Import and export license	17	Pharmacy license
6	Trademark registration	18	Private educational institution license
7	Factory license	19	Adult education
8	Auditing firm license	20	Tourist guide license
9	Media house registration	21	Travel agency license
10	Marine operations and works license	22	Bank license
11	Marine taxi license	23	Money exchange agent license
12	Food and animal origin license	24	Remittance company license

Electronic Government

The use of Information and Communication Technology (ICT) to transform the internal workings of government and its relationships with citizens, businesses and other arms of government.

1	Focuses on improving every aspects of public service delivery: o Structural transformation of government agencies (rules, norms, roles, responsibilities) o Review and modification to business processes, with improved technology support o Enhancing access to services through various channels
2	Also aims at providing personalized information to stakeholders to meet their needs and enable participation in government decision processes.

Electronic Public Services (EPS)

Public services delivered over electronic channels such as the Internet, telephone, mobile devices, kiosks, digital TV, etc.

1	The number and sophistication of available EPS is one of the main indicators for the maturity of Electronic Government: 1) Information Services 2) Interactive Services 3) Transactional Services 4) Seamless Services 5) Personalized Services
2	Scaling up the provision of EPS is generally challenging

A technical focus for Electronic Government is the comprehensive provision of EPS.

Outline

EPS Infrastructure

A software infrastructure which supports the development and operation of Electronic Public Services by providing shared or common components and services.
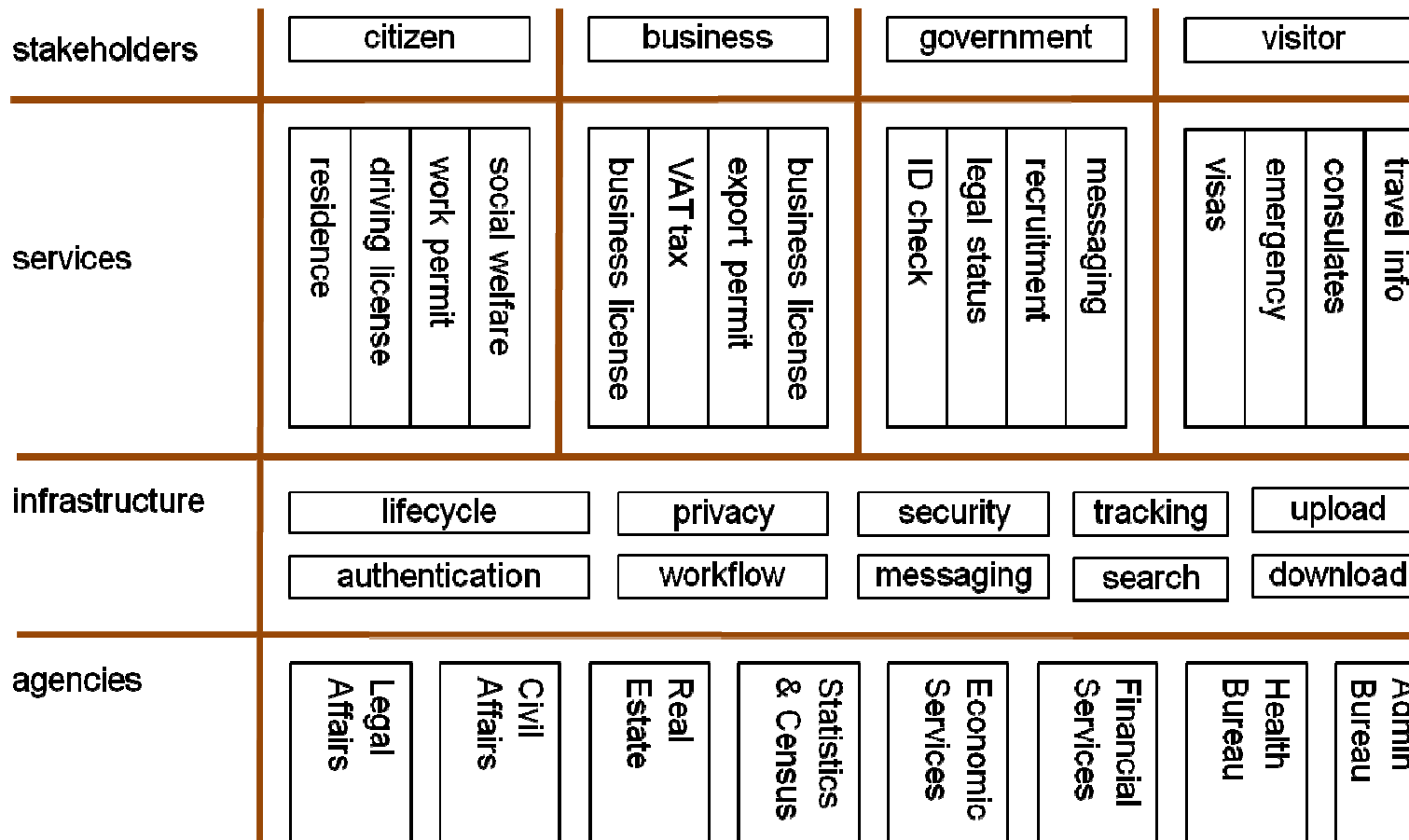
EPS Infrastructure - Rationale

Why Software Infrastructure for EPS?	
Scaling	Rapid development of EPS through composition and reuse of shared infrastructure elements.
Standardization	Streamlining government practices, for instance through the use of generic e-forms, authentication services and administrative processes.
Control	Basic control can be effected through the infrastructure by exploiting the information generated about services through transaction logs, etc.
Cost sharing	By polling the resources to develop shared components and services, agencies can reduce the EPS development costs significantly.

EPS Infrastructure - Development

Most government understand the need for an EPS infrastructure in the provision of mature EPS. Unfortunately, there is lack of frameworks/guidelines on how to develop such an infrastructure.

A major focus for UNU-IIST Center for Electronic Governance is providing such know-how by:

1	Capturing domain knowledge through concrete practice experience
2	Developing domain models for EPS and validating such models
3	Providing prototype implementation of EPS infrastructure based on developed models
4	Assisting governments in transforming Prototype to Production-Quality EPS Infrastructure
5	Continued enhancement of EPS infrastructure models and dissemination of results

EPS Infrastructure - Functional Requirements

As a minimum, support authorization-type of EPS.

NO	USE CASE	ACTOR
1	Submit Application	Applicant
2	Upload Supporting Documents	Applicant
3	Check Validity and Completeness	Public Agency
4	Verify Evidences	Public Agency
5	Assess Eligibility	Public Agency
6	Decide On Application	Public Agency
7	Notify Applicant	Public Agency
8	Track Application	Applicant

EPS Infrastructure - Non-Functional Requirements

1	Explicit support for technical and organizational interoperability
2	Built on open standards and technologies
3	Explicitly supporting customization and localization - primarily purpose for development is dissemination in different environments
4	Conceptually simple, but powerful and easy to evolve
5	Well documented interfaces for support by third-party software houses
6	Implemented within an defined IT Governance Framework (standards)

The EPS Infrastructure

The EPS Infrastructure provides design-time elements to enable rapid development of EPS, and run-time elements to support the execution and operation of EPS.
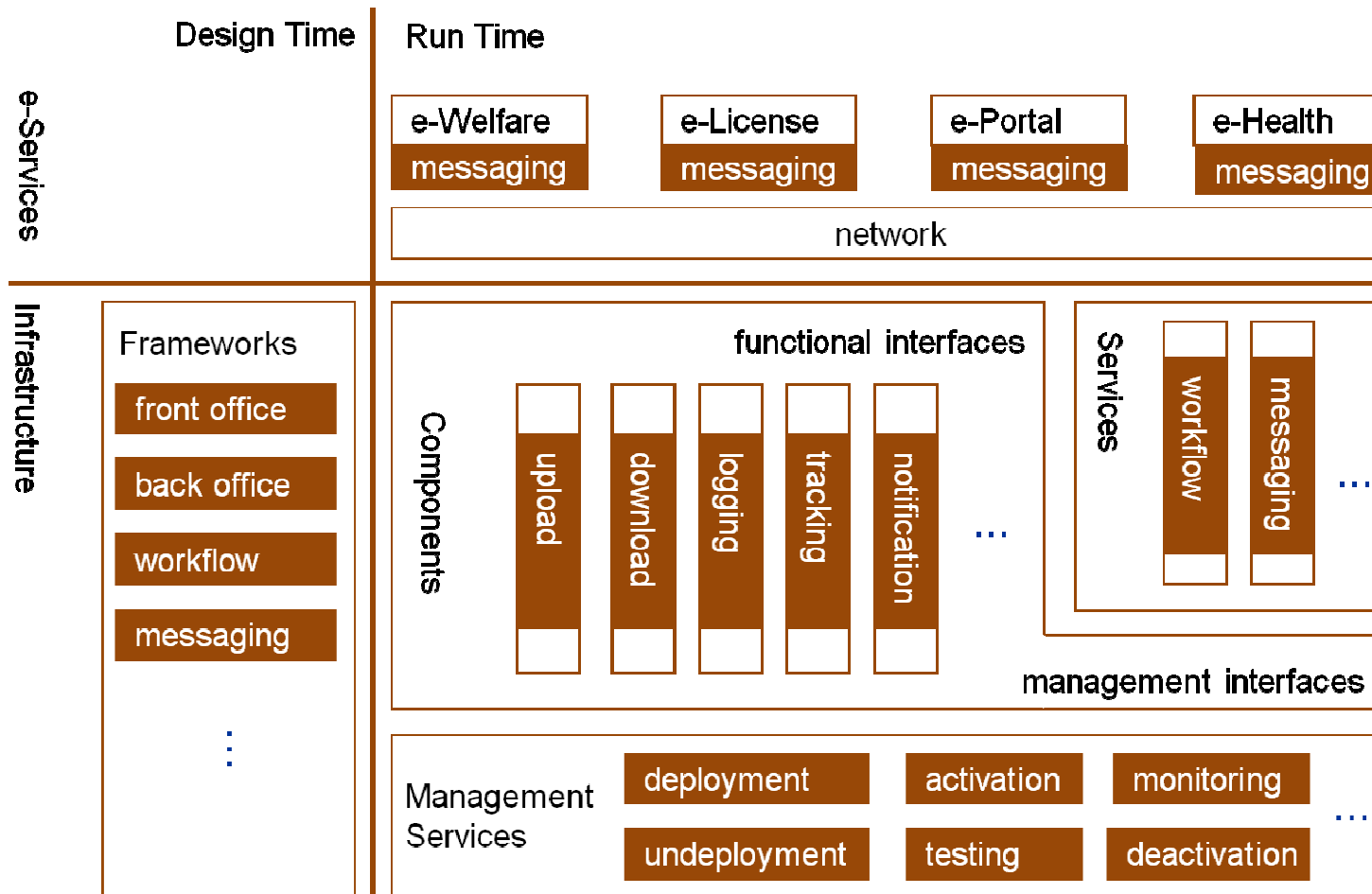
Element 1 – Front Office Framework

Building Front-Office parts of EPS with the following features:	
1	EPS receives requests from client applications
2	EPS validates all received requests from clients
3	EPS generates request receipts for applicants
4	EPS supports tracking of applications by applicants
5	EPS dispatches requests to the workflow service

Element 2 – Back Office Framework

Building Back-Office parts of EPS with the following features:	
1	EPS checks completeness of submitted documents
2	EPS assesses eligibility of applicants
3	EPS supports tracking of applications by Back-Office officers
4	EPS notifies applicants over different channels
5	EPS supports evaluation and decision steps
6	EPS exchanges messages with other Back-Offices through the Messaging Service

Element 3 – Workflow Service

Build Mid-Office parts of EPS to enable coordination:	
1	EPS receives requests from the Front-Office applications
2	EPS associates requests with business processes
3	EPS creates Back-Office tasks for execution
4	EPS assigns Back-Office roles to tasks and dispatches tasks to the Back-Office
5	EPS receives task completion tasks from the Back-Office

Element 4 – Messaging Service

Build Mid-Office parts of EPS to enable data exchange:	
1	EPS can exchange messages along dynamically-created and subscribed channels
2	EPS can carry out asynchronous communication between Back-Offices in different agencies
3	EPS can rely on horizontal extensions to channels, e.g. transformation, validation, security.
4	EPS can rely on vertical extensions for channels, e.g. semantics, process enforcement

Element 5 – Management Service

Manage the lifecycle of other infrastructure elements:	
1	Registers all infrastructure elements and assigns them unique IDs
2	Logs the actions of other components and services
3	Starts-up, shuts down and suspends the operations of services
4	Controls the behavior of other elements
5	Allows for service- or component-level management

Outline

EPS Development

EPS development relies on the use of all five infrastructure elements, as follows:	
1	Developing Front- and Back-Office parts of EPS using the corresponding frameworks
2	Developing workflow systems to connect Front- and Back-Office parts of EPS
3	Connecting different Back-Offices through the Messaging service
4	Configuring the three major subsystems to be individually manageable

EPS Development - Structure
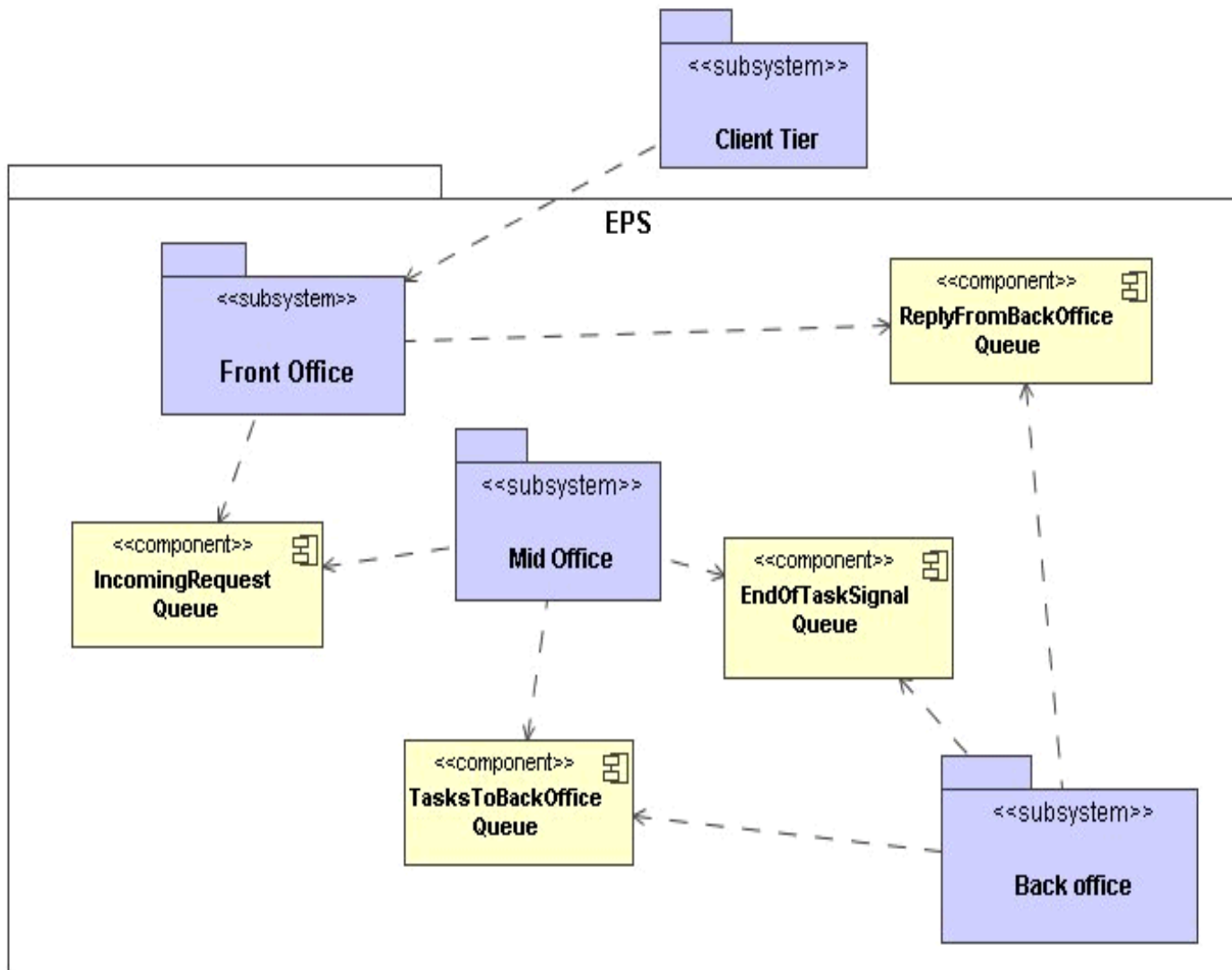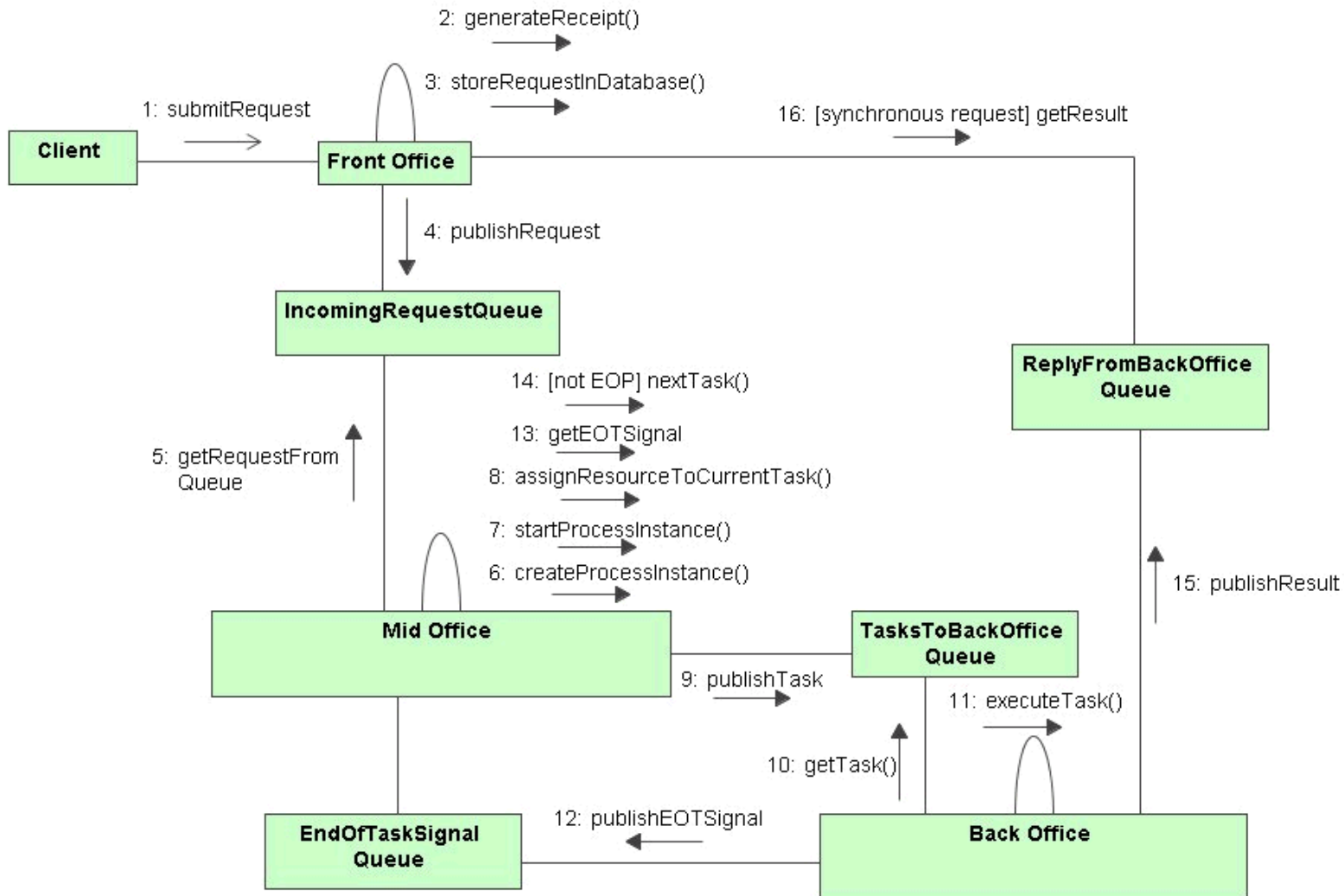
EPS Development - Process
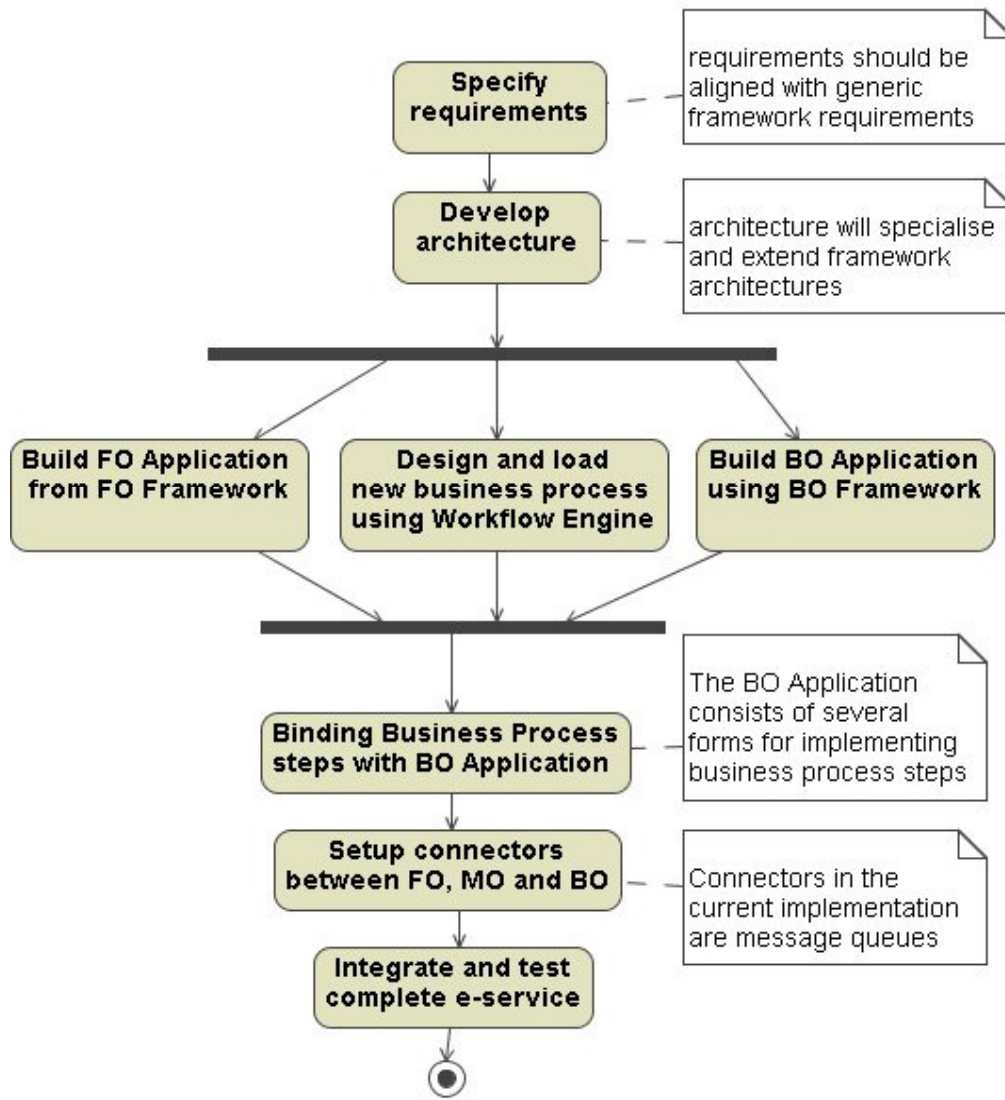
EPS application engineering process:	
1	Specify requirements
2	Develop architecture
3	Build FO part using FO Framework
4	Design and load new business process using the Workflow Service
5	Build BO part using BO Framework
6	Bind business process steps with BO
7	Connect FO, MO and BO using queues
8	Integrate and test EPS

Outline

Trustworthy EPS Infrastructure

As many EPS are built from infrastructure elements and also require these elements at run time for services, the trustworthiness of all infrastructure elements is essential.

Trustworthiness here entails:

1) availability,
2) reliability,
3) security,
4) responsiveness and
5) performance

of all infrastructure elements (frameworks, components and services).

One way to realize a Trustworthy EPS Infrastructure is to ensure that all infrastructure elements are well managed.

Manageable EPS Infrastructure

We refer to the management of web resources as captured by three OASIS specifications:	
1	Web Service Distributed Management (WSDM)
	A base specification for managing any kind of IT resource based on Web Services. It defines a set of properties to describe and capture the state of any kind of IT resource, enabling easy integration of the management aspects of IT resources in general.
2	WSDM - Management Using Web Services (WSDM - MUWS)
	Defines how an IT resource connected to a network provides manageability interfaces to be managed both locally and from remote locations using Web Services technologies.
3	WSDM – Management of Web Services (WSDM – MOWS)
	Extends WSDM-MUWS particularly for the management of Web Services.

Contracts

With managed infrastructure elements, information on the history and state of infrastructure elements is available through various resource property values.

Based on these properties, a contract can be defined between an infrastructure element and a consumer entity (other infrastructure element or external application) to guarantee some degree of trust between interacting entities.

Our notion of contract is a formal relationship between two or more parties that use or provide resources where rights, obligations and negotiation rules over resources, expressed as:

1) Pre-conditions
2) Post-conditions
3) Invariants
4) Protocols
5) Quality of Service requirements
6) Etc.

QoS Contracts

We focus on QoS contracts - non-functional aspects of a service such as performance, reliability, availability, security, associated with specialized functions provided by the service.

Different QoS parameters for different kinds of services:

Task	deadline, response time, criticality, priority, availability, accountability
Messaging	delivery guarantee, duplicate elimination, ordering, message delay, confidentiality, integrity and retry limit
Streaming	throughput, jitter, accuracy

We are more interested in Task and Messaging services.

QoS Contract Specification

A number of contract specification languages, all based on XML and relatively informal:	
1	Web Service Level Agreement (WSLA)
	Allows to express: what to measure, how to measure, who does what and guarantees. It is an XML based language for specifying SLA-related information
2	Quality of Service Modeling Language (QML)
	A general purpose QoS specification language not tied to any particular domain such as real-time or multi-media system or to any particular QoS category.
3	User-Defined Languages
	Simply define a set of parameters and identifies the parties involved in the roles of provider and consumers.

Adopted QoS Specification

We adopt a generic QoS specification language similar to Wang et. al 2007, but focusing on the task and messaging QoS parameters due to the nature of infrastructure elements.

QoS parameter types and dimensions:

Type	• task – Front-Office Framework, Back-Office Framework, Workflow Service • messaging – Messaging Service
Dimensions	• numeric • ordinal (e.g. Low, Medium, High)

Adopted QoS Specification Example
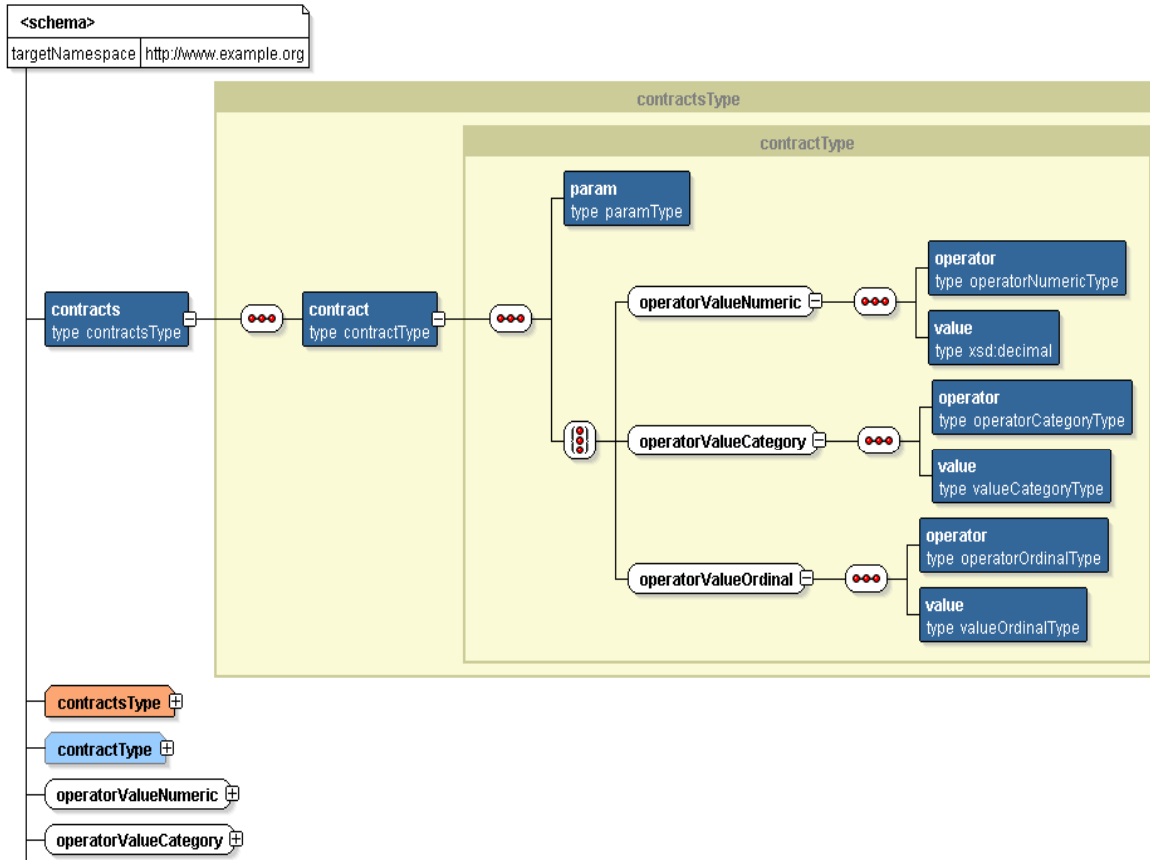
QoS Contract Schema fragment:

```
…
<xsd:element name="contracts"
type="contractsType"/>

  <xsd:complexType name="contractsType">
    <xsd:sequence>
      <xsd:element name="contract"
type="contractType"/>
    </xsd:sequence>
  </xsd:complexType>


  <xsd:complexType name="contractType">
    <xsd:sequence>
      <xsd:element name="param"
type="paramType"/>
      <xsd:choice>
        <xsd:group ref="operatorValueNumeric"/>
        <xsd:group
ref="operatorValueCategory"/>
        <xsd:group ref="operatorValueOrdinal"/>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
…
```

Contract Monitoring

Contract monitoring is achieved by capturing information during execution of infrastructure services and components.

1	Each infrastructure element is wrapped with a web service for manageability
2	The web service wrapper is implemented as an intermediary or handler to intercept calls to infrastructure elements and replies from them
3	The intermediary updates the resource property documents based on calls to and replies from infrastructure services
4	The intermediary provides operations to deliver QoS parameters to requesting consumers

Contract Verification

Contract verification between producer and consumer elements is carried out by comparing parameter values in contracts with actual parameters values recorded in resource property files.

Verification steps:

1	Contract verifier contacts management intermediary for specific QoS-related information
2	The verifier compares values in the contract with the values in the resource property file
3	Mediation action is activated based on the results of comparison

Contract Mediation

Mediation action is aimed at enforcing producers to deliver services at the agreed level specified in the contract.

Possible mediation actions:

1	Specific mediation actions would depend on nature of offending parameters.
2	General mediation actions can halting, suspending, reducing granularity of logging, etc.
3	Re-negotiation of contract is a possible response.

Contract Management Implementation - IMS

1	Partial contract management implementation is provided by the EPS infrastructure through the Infrastructure Management Service (IMS)
2	IMS creates an extended WSDM-MUWS and WSDM-MOWS resource property files for each infrastructure element during deployment on the infrastructure
3	Contract monitoring is done through the IMS Handler provided with each infrastructure element at deployment
4	IMS basic services provides QoS information on demand
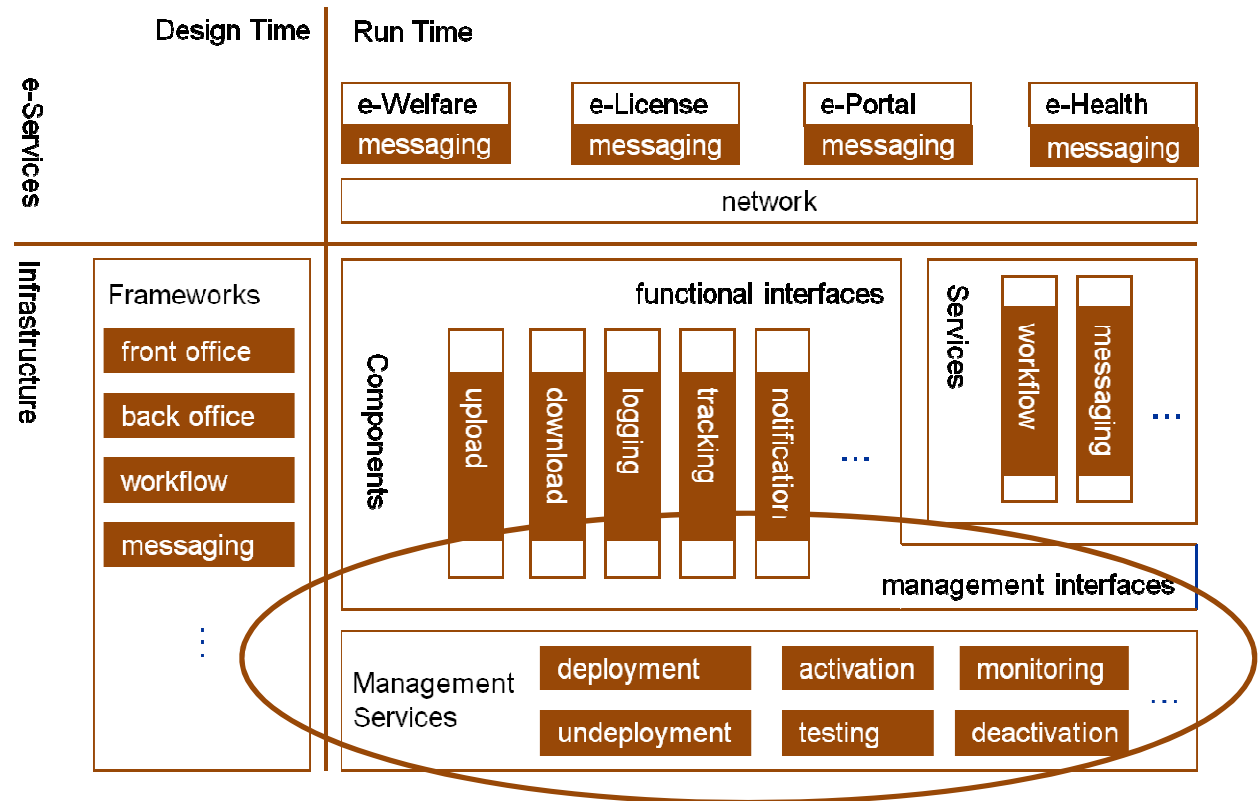
Example – IMS Monitoring Service

Shows the contents of a resource property file for one of the infrastructure service.

Example – IMS Mediation Actions

Shows the general mediation or control services for IMS – start, stop, resume, suspend.

Outline

Conclusions

1	The goal of the work is to develop a trusted EPS infrastructure that could impact, on the long run, on the citizen's trust in the delivery of public services
2	Our approach is to ensure the manageability of infrastructure elements by providing management capabilities within the EPS infrastructure
3	To guarantee a certain level of trust with respect to the EPS infrastructure, contracts are defined between consumer entities and infrastructure elements as service providers
4	Simple XML-based language is defined to specify QoS parameters and allowed values
5	QoS contract management is implemented partially through IMS - Infrastructure Management Service - which provides overall lifecycle management functionality for infrastructure elements.
6	Future work entails defining a comprehensive QoS contract management capabilities which is directly linked to higher level business processes underpinning Electronic Public Services.

Bibliography 1

1) Slung-Yong and Lung-Teng Hu, Citizen's trust in Digital Government, Graduate Department of Pubic Admin, Rutgers, the State University of New Jersey.

2) Chrisanthi Avgerou, Andrea Ganzaroli, Angeliki Poulymenakou, Nicolau Reinhard, ICT and Citizen's trust in government: lessons from electronic voting in Brazil, Proceedings of the 9th Conference on Social Implications of Computers in Developing Countries, Sao Paolo, 2007

3) Bill Gates: Trustworthy Computing, [online] available at http://www.wired.com/techbiz /media/news/2002/01/49826

4) Ojo, A., Janowski, T., Oteniya, G., Estevez, E., Infrastructure Support for e-Government – An Overview, e-Macao Report 6, Aug. 2006.

5) Ojo, A., Chu, T. I., Oteniya, G., Tou, C. P., Janowski, T., Front-Office Framework for e-Government, e-Macao Report 8, Aug. 2006.

6) Estevez, E., Wan, C., Wong, C. T., Ojo, A., Oteniya, G., and Janowski, T., Back Office Framework for e-Government, e-Macao Report 9, Aug. 2006

Bibliography 2

7) Estevez, E., and Janowski, T., Extensible Message Gateway for e-Government, e-Macao Report 11, Aug. 2006.

8) Joe, A., Oteniya, G., and Janowski, T., Infrastructure Management Services for e-Government, e-Macao Project Report 12, Aug. 2006.

9) An Introduction to WSDM, Committee Draft, OASIS, Available online http://www.oasis-open.org/committees/download.php/16998/wsdm-1.0-intro-primer-cd-01.doc, Feb. 2006.

10) Web Services Distributed Management: MUWS Primer, Committee Draft, OASIS, Available online http://www.oasis-open.org/committees/download.php/17000/wsdm-1.0-muws-primer-cd-01.doc, February 24, 2006

11) Web Services Distributed Management: MOWS Primer, Committee Draft, OASIS, Available online http://www.oasis-open.org/committees/download.php/17001/wsdm-1.0-mows-primer-cd-01.doc, February 24, 2006.

12) Web Services Management: Service Life Cycle, W3C Working Group Note, Available online at http://www.w3.org/TR/wslc/11 February 2004.

Bibliography 3

13) Struts in Action, Ted Husted, Cedric Dumoulin, George Franciscus, David Winterfeldt, Manning Greenwich, 2003.

14) Orlando Loques and Alexandre Sztajnberg, Customizing Component-Based Architectures by Contract, Component Deployment, pp 18-34, 2004.Svend Frølund and Jari Koistinen, Quality of services specification in distributed object systems design. In Proceedings of the 4th Conference on USENIX Conference on Object-Oriented Technologies and Systems - Volume 4 (Santa Fe, New Mexico, April 27 - 30, 1998). Conference on Object-Oriented Technology and Systems. USENIX Association, Berkeley, CA, 1-1, 1998.

15) Changzhou Wang, Guijun Wang, Haiqin Wang, Alice Chen, and Rodolfo Santiago, Quality of Service Contract Specification, Establishment, and Monitoring for Service Level Management, Vol. 6, No. 11, Special Issue on Advances in Quality of Service Management, December 2007.

16) IBM, Web Service Level Agreements (WSLA) Project, http://www.research.ibm.com/wsla/