A Security Contextualisation Framework for Digital Long-Term Preservation

Kun Qian[1], Maik Schott[1], Christian Kraetzer[1], Matthias Hemmje[2], Holger Brocks[2], Jana Dittmann[1]

[1] Faculty of Computer Science, Otto von Guericke University Magdeburg, Germany
[2] Faculty of Mathematics and Computer Science, University of Hagen, Germany

Acknowledgment
The work in this paper has been supported in part by the European Commission through the FP7 ICT Programme under Contract FP7-ICT-216736 SHAMAN. The information in this document is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

Overview

- Introduction and Motivation
- Design of the Framework
- Exemplary Application
- Summary

Introduction and Motivation

- Ever increasing amount of born-digital data
- Cultural heritage **à** long-term (i.e. min. 100 years) preservation is necessary
- Company documents **à** statutory period of record keeping
- Standard of preservation systems: OAIS reference model

Introduction and Motivation

- Security only insufficiently addressed
  - Security aspects are sparsely mentioned but no requirements are given
  - No mention of security policies
  - Extension of OAIS necessary

- Coverage of all security aspects

Overview

- Introduction and Motivation
- Design of the Framework
- Exemplary Application
- Summary

Design of the Framework

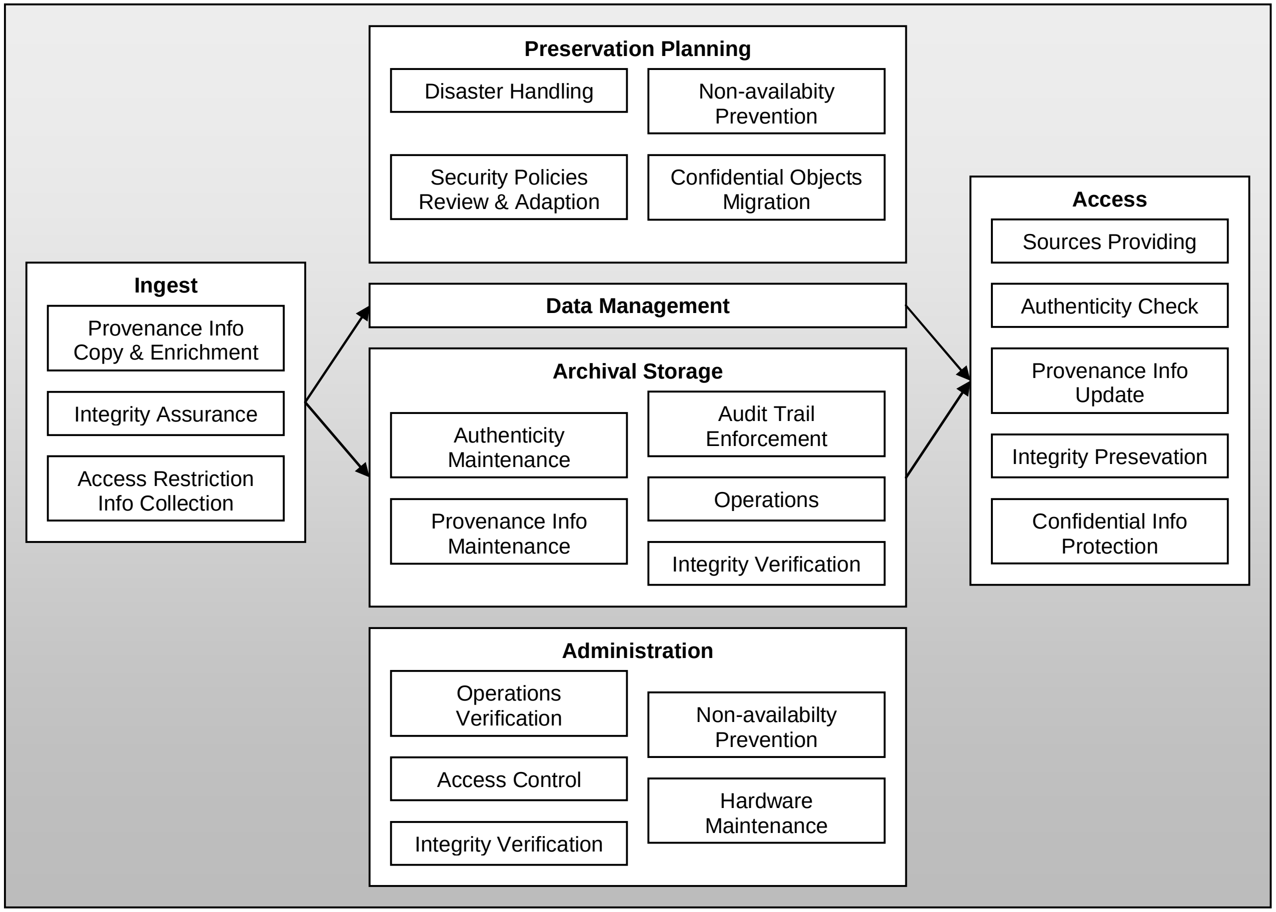
Major functional blocks

- Context modelling
- Policy generation hierarchy
- Information Package (IP) processing
- Control

Design of the Framework

Context modelling

- Ontology based context modelling
- Extending OAIS standard with security-related functions



Design of the Framework

Policy generation

- Specification of policy requirements
  - Identification and classification of security objects, subjects, and their relations
  - Meta policies guarantee that these requirements are basic properties of the security policies

Design of the Framework

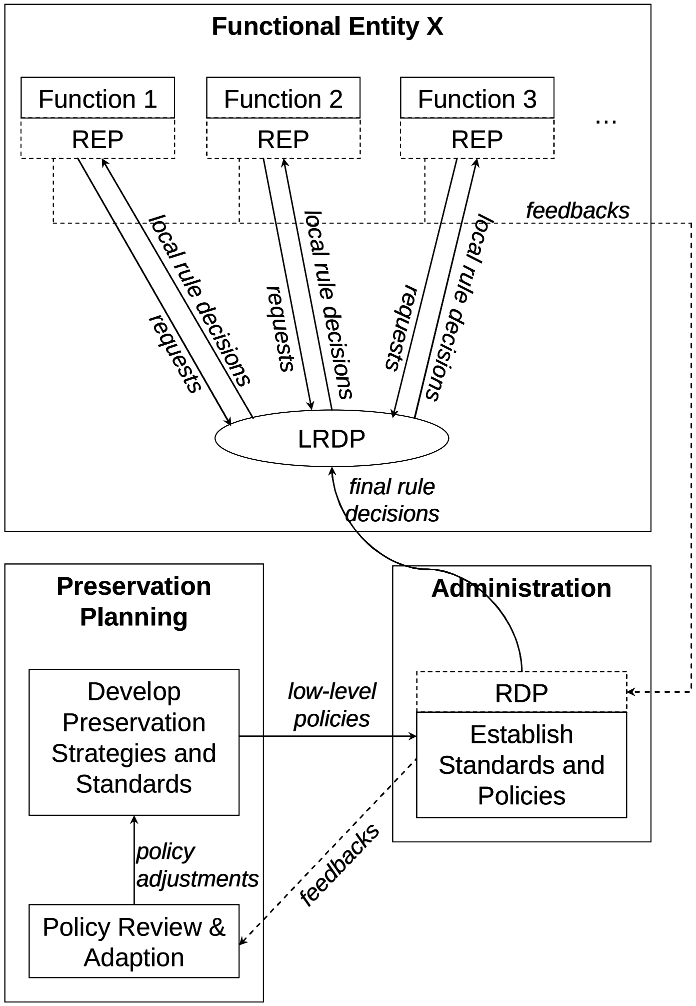
Policy generation

- Policy derived from use-cases
- Policy managed using a four level hierarchy
  - Meta policies
  - high-level policies (overall requirements)
  - mid-level policies (module requirements)
  - low-level policies (concrete actions)
- Policy identifiers applied to ensure clarity and traceability

Design of the Framework

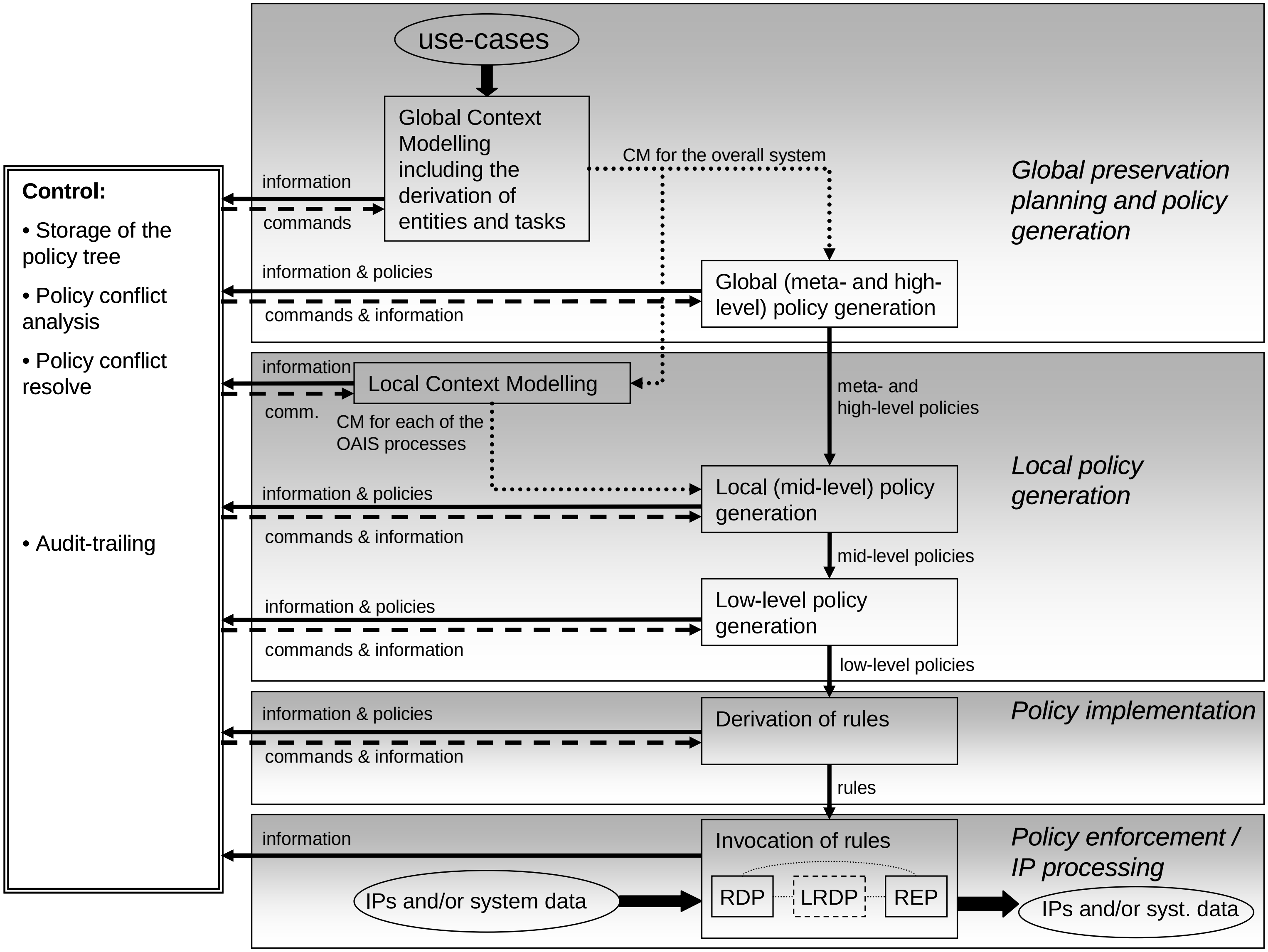
Policy implementation and enforcement

- Implementation: low-level policy à enforceable rule
  - Create rules
  - Instantiate rules
  - Validate rules
- Enforcement: execute rules by adapting RFC2753 and COPS
  - Policy Decision Point
  - Policy Enforcement Point



Design of the Framework

General overview of the framework



use-cases

Global Context Modelling including the derivation of entities and tasks

CM for the overall system

information

commands

Global preservation planning and policy generation

information & policies

commands & information

Global (meta- and high-level) policy generation

information

Local Context Modelling

comm.

CM for each of the OAIS processes

meta- and high-level policies

Local policy generation

information & policies

commands & information

Local (mid-level) policy generation

mid-level policies

information & policies

commands & information

Low-level policy generation

low-level policies

information & policies

commands & information

Derivation of rules

Policy implementation

rules

information

Invocation of rules

Policy enforcement / IP processing

IPs and/or system data

RDP

LRDP

REP

IPs and/or syst. data

Control:

- Storage of the policy tree
- Policy conflict analysis
- Policy conflict resolve
- Audit-trailing

Overview

- Introduction and Motivation
- Design of the Framework
- Exemplary Application
- Summary

Exemplary Application

Context model - functional entity *ingest*

- Receive submission
- Quality assurance
- Generate AIP
- Generate descriptive info
- Coordinate updates

- Provenance info copy and enrichment
- Integrity assurance
- Access restriction info collection

Exemplary Application

High-level Policies

Policy	**Description**
P01	The repository provides the means to authenticate all objects by providing/identifying the sources it was created from.
P02	The repository ensures authenticity of digital objects for all steps of processing.
P03	The operations must be logged including what the operation has processed, on whose behalf, when and with which result.
P04	To ensure the correct operation of mechanisms, it must be validated/verified that they correspond to the defined policies.
P05	The system provides mechanisms to authenticate subjects.
P06	The integrity of objects must be guaranteed for all processing steps.
P07	The repository's integrity must be preserved.
P08	The actual performance of the security preservation mechanisms must be audited by an independent party.
P09	The audit trail must be available at any time; conversely the system must not operate without an audit trail although this will result in non-availability.
P10	Confidential information must not be disclosed.

Exemplary Application

Selected mid-level policies

Policy	Description
P02-01	If the SIP are ingested and thus becoming one or more AIPs the authenticity/provenance must also be preserved, by including the SIP provenance information and enriching them with data about the ingest (authenticated producer, time, etc.).
P02-02	Necessary changes to be made to the content of SIPs to receive a valid AIP must be stated in the AIP provenance info.
P06-01	If SIP are ingested and thus becoming one or more AIP the integrity must be preserved which especially includes semantic and referential integrity and that the SIP are ingested as AIP completely.
P06-02	If objects are migrated, for example by conversion into a new format, the integrity of the old and the new version must both be enforced, and the newer version should be at least semantic integer with the older version directly after the conversion.
P07-01	As the ingest is the major part where external data is fed into the system, special means must be taken that these do not compromise the security of the archive.
P07-02	Although the ingestion should be as fault tolerant as possible, SIP must be ingested either wholly/completely or not at all, but never partially.
P10-01	If the SIP are ingested and thus become one or more AIP, the corresponding AIP must fulfill the defined confidentiality conditions and generate appropriate access restrictions.

Exemplary Application

Selected rules corresponded to low-level policies

Rule	Func.	Description

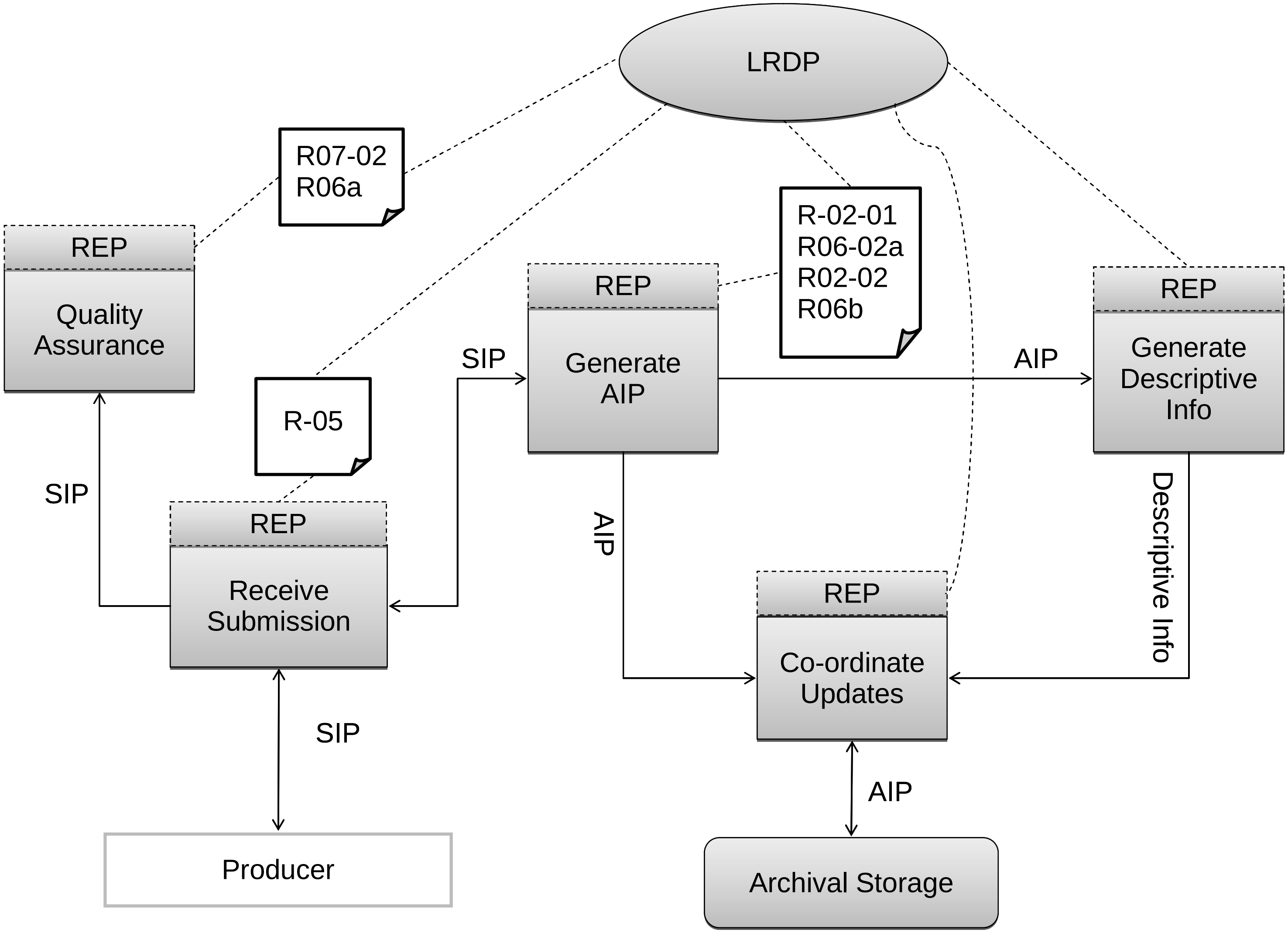
R02-01	Generate AIP	AIP.PDI = SIP.PDI
		AIP.PDI.ProvenanceInfo.Add(Time, Producer, “Ingested from ” + AIP);

		If ValidArchivalFormat(SIP.Content, ArchivalFormats) then AIP.ContentInfo = SIP.ContentInfo; else
R06-02a		AIP.ContentInfo = Convert(SIP.ContentInfo); text = “Format conversion from %s to format %s”); Log(AIP, text);
R02-02		AIP.PDI.ProvenanceInfo.Add(Time, Text);
R06b		AIP.PDI.FixityInfo.Hash = Hash(AIP. Content);
		Endif;
R10		If AIP.IsConfidential() then AIP.Content = Encrypt(AIP, Content); Log(AIP, “Encrypted”); Endif;

R06-01		If AIP.PDI.ContainsSIPReferences() then AIPQueue.Put(AIP); IngestNextSIP();Endif;
		If NoNextSIP() ReplaceSIPReferencesByAIPReferences(AIPQueue); Endif;

Exemplary Application

Rule enforcement



Overview

- Introduction and Motivation
- Design of the Framework
- Exemplary Application
- Summary

Summary

- A contextualisation framework
  - Application scenario: digital long-term preservation
  - Bottom-up approach for context modeling
  - Derivation of hierarchical policies from use-cases
  - Policy implementation and enforcement through rules

Questions?