### 1. BACKGROUND

- Cloud application platforms will revolutionize the development and delivery of software applications.
- By design, a cloud application platform is an open environment that is meant to expand over time.
- Maintaining platform integrity and reliability in face of continuous expansion is a major challenge for the platform provider.
- Definition and enforcement of policies to govern platform processes and artefacts is essential.

### 2. TYPES OF POLICY-DRIVEN GOVERNANCE

- **Process governance:**
  - The evolution of managed entities must follow an explicitly defined lifecycle model consisting of specific states and transitions, where transitions are guarded by preconditions.
  - Example: An app can proceed to beta testing only if there exists an associated QA staff review report with a positive evaluation.

- **Artefact governance:**
  - Artefacts associated with managed entities must conform to all technical or business constraints set by the platform provider.
  - Example: The interface specification (WSDL) of every external web service should contain exactly two non-identical endpoint URLs (primary and backup servers for failover).

### 3. PROBLEM DESCRIPTION

- Limitations of policy handling in existing governance tools (registry & repository systems)
  - Lack of proper abstraction: Policy logic is represented at the same level of abstraction as the implementation logic of the R&R system.
  - Lack of separation of concerns: Policy definition and enforcement entangled in the same piece of code.
  - Lack of formal representation of the relationships among policies and between policies and their subjects (i.e. the logical entities in the domain of governance).

- Implications:
  - Limited policy traceability, maintainability, comprehensibility, verifiability, interoperability, and overall governance agility.

### 4. GOALS & OBJECTIVES

- **Main thesis of this PhD research:**
  - The limitations of governance technology can be addressed by an ontology-driven approach to defining and enforcing policies.

- **Objectives:**
  - Development of a cloud platform governance ontology and a policy modelling methodology based on our CAST project dataset.
  - Development of a policy conformance checking engine supporting alternative methods of policy analysis.
  - Definition of a framework architecture enabling integration into various governance registry and repository systems.

### 5. APPROACH OUTLINE

- Two alternative strategies for ontology-based policy definition and enforcement:
  - Definition of policies as OWL DL class axioms, data conformance checking via OWL DL reasoner’s instance checking service.
  - Definition of policies as SPARQL queries, data conformance checking via query answering.

### 6. RESULTS TO DATE (ONGOING WORK)

- Two alternative strategies for ontology-based policy definition and enforcement:
  - Definition of policies as OWL DL class axioms, data conformance checking via OWL-DL reasoner’s instance checking service.
  - Definition of policies as SPARQL queries, data conformance checking via query answering.

- Present state of cloud platform governance ontology (adopting first strategy):
  - Language: OWL 2; Expressivity: ALCIQ(D) plus SWRL; Size: 170 classes, 30 properties, 29 individuals (constants), 7 SWRL rules; OWL 2 language features: XSD facets, Keys.

- Challenge: standard OWL semantics (Open World Assumption and no Unique Name Assumption); instance data validation achievable through preprocessing step of closure axiom generation.