

# Use-Cases / Technology Session

DE Cluster Meeting, Brussels  
21-22<sup>nd</sup> May, 2007



# Agenda

---



- Introduction (Luigi Telesca and Javier Vázquez)
- Individual Presentations:
  - SEAMLESS Project (Juanvi Vidagany, Radoslav Delina)
  - e-NVISION Project (Valentín Sánchez and Marek Tarka)
  - VISP (Nicolas Fonrose)
  - Coffee Break*
  - ONE Project (Luigi Telesca)
  - CONTRACT Project (Javier Vázquez)
- Discussion on Use-Cases / Application Areas / Technical Approaches

# IST CONTRACT Project

IST CONTRACT Project  
FP6



# IST Contract Project



- IST FP6 STREP Project starting 1<sup>st</sup> Sept 2006
- Focus:
  - Contracts for Distributed Applications Engineering
  - Contracts as a basis for formal verification
  - e-business applications
- Non-focuses:
  - Negotiation, capturing human contracts

**KING'S**  
*College*  
**LONDON**

University of London



**FUJITSU**

**Imperial College**  
London



Certicon A. S.

Yall B. V.

LostWax Media Ltd.

# CONTRACT in a NUTSHELL?



- “Engineering applications in Cross Organisational Service Oriented Computing environments”
- Trying to make progress on three fronts
  - Build on the idea that **formal verification over *contracts*, obligations etc.** rather than over internal code is the way to build sound distributed applications in service oriented environments.
  - Create **concrete methods and tools** which enable the use of contracts, obligations and agreements in order to structure the design of such applications (target audience WS-\* community)
  - Build realistic **use cases** in different industries
- Focus on Contracts
  - As the explicit, tangible representation of service interdependencies

# What do we mean?

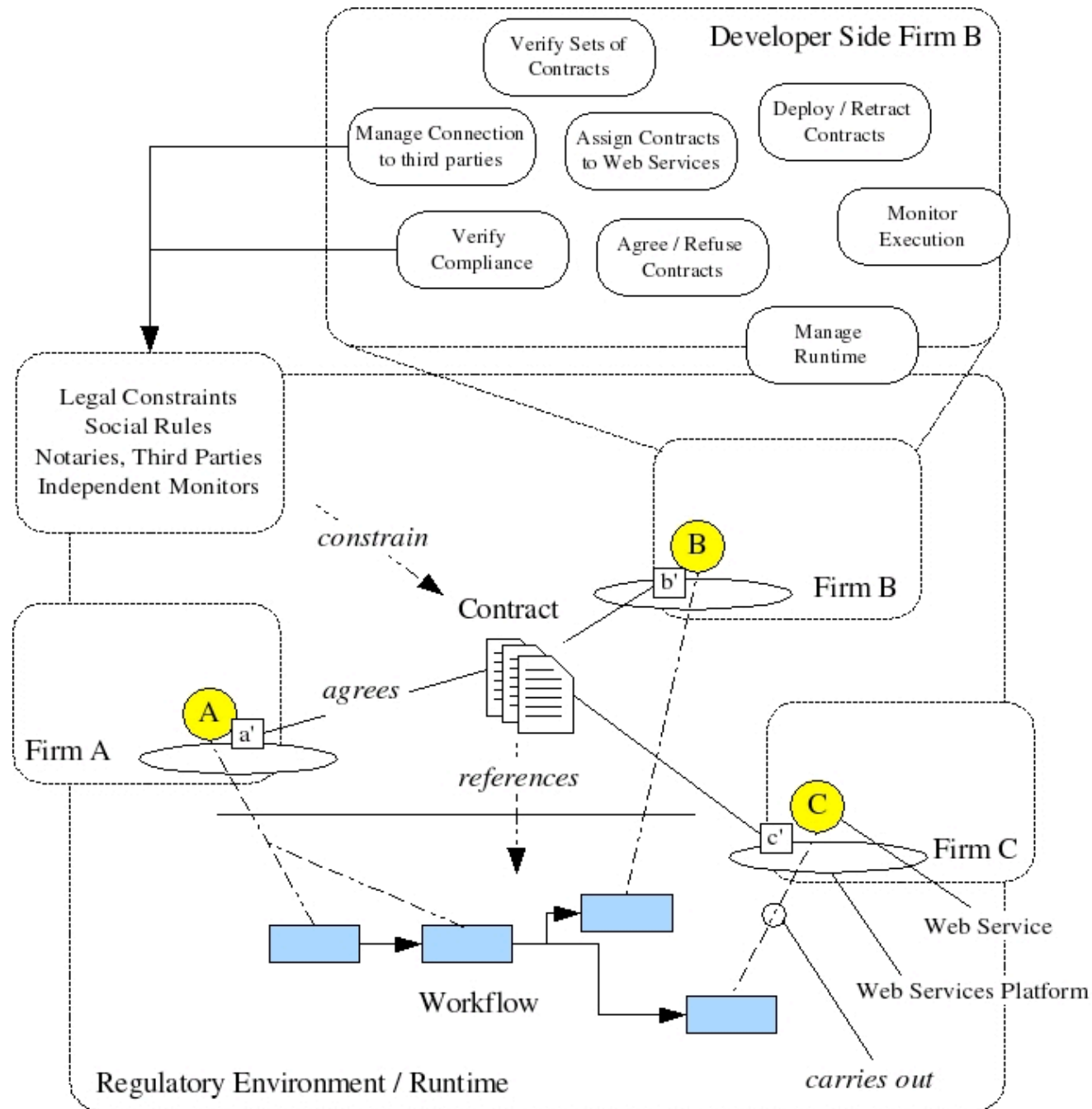


- The behaviour of a software application depends upon:
  - Code
  - Execution Context (environment)
  - Inputs
- In a multi-organisational Web Services application:
  - No-one has access to all the code
  - No-one has access to all the execution context
  - (Possibly) no-one has access to all inputs
- Question:
  - How do you predict the potential run-time behaviour of such applications?

# Project Core Idea



- Formal Verification approaches for software will not work without this access
- Contract Project Approach:
  - **Exchange Access to Code / Environment for Access to Contracts**
- Instead of predicting actions w.r.t code - predict actions w.r.t obligations, rights, permissions
- Impact:
  - Public information, (in principle) massively reduced search space, public impact of failure

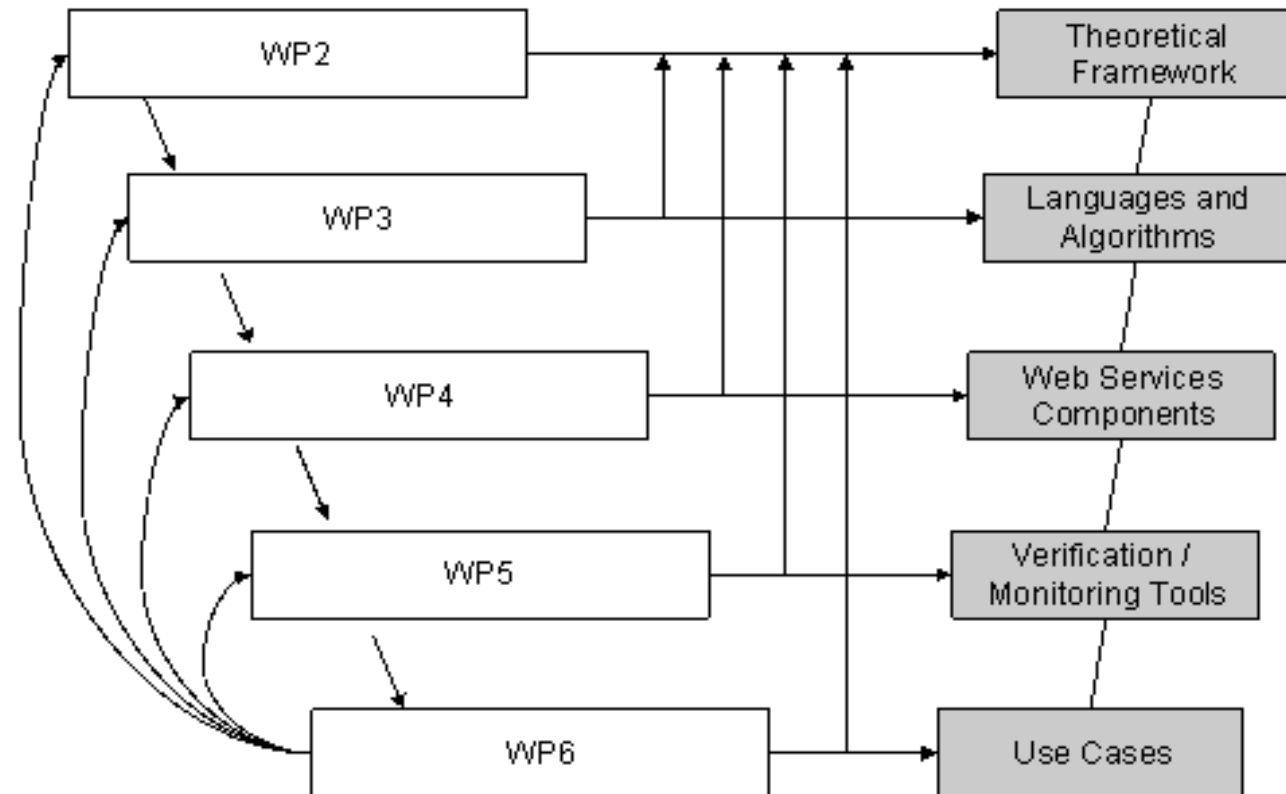


# Project Target Outcomes



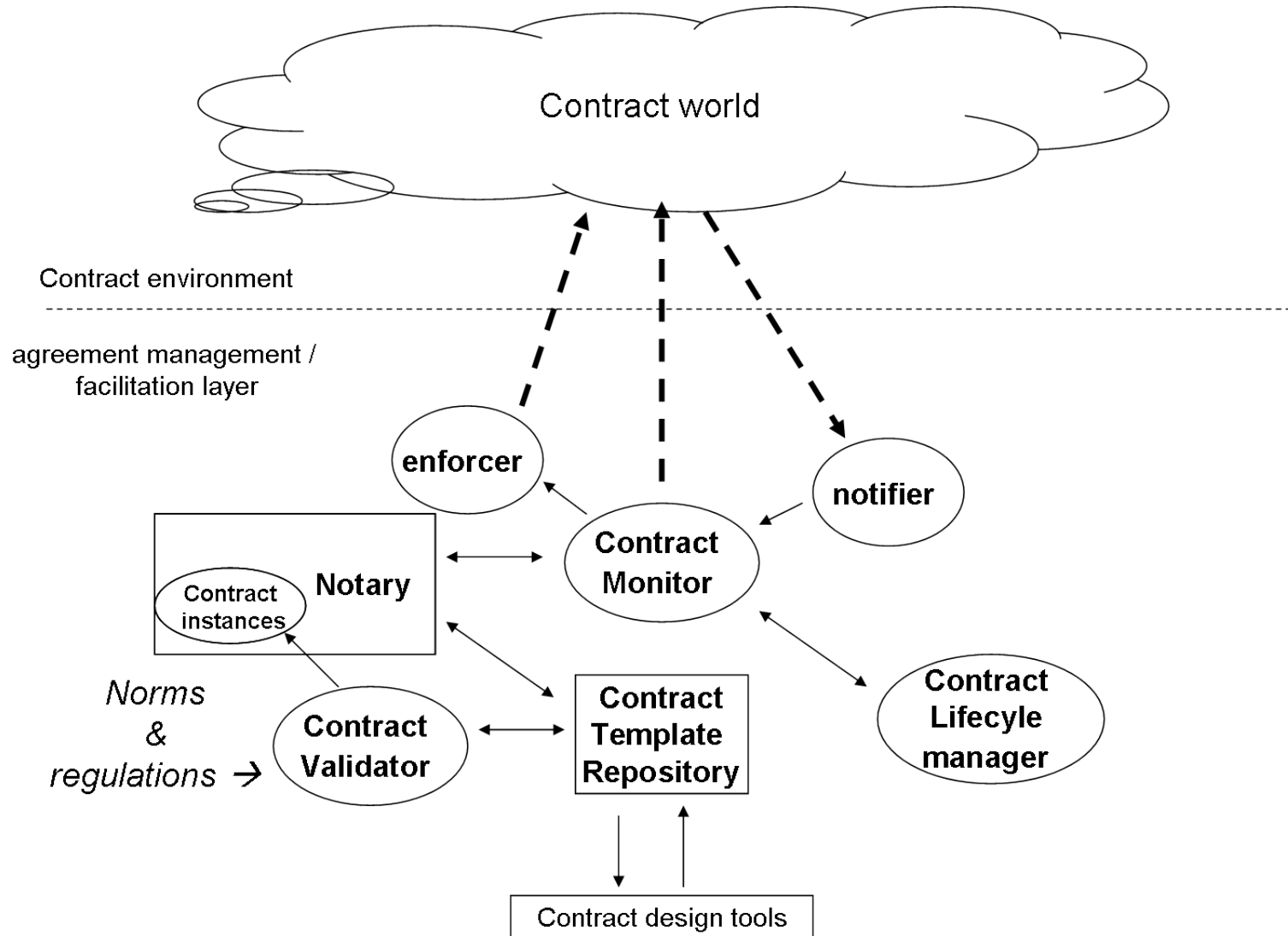
- Major Outcomes:
  - Theoretical Framework
  - XML/RDF Based Contracting Language Syntax and Semantics plus associated verification techniques
  - Contract based e-Business Web Services Application Frameworks
  - Verification, monitoring and analysis tools
  - Three case studies
- Timeline:
  - Specifications / Document from Months 9-12...
  - Software from Months 15-18...
- Majority of results open, royalty free, open source

# Major Work Areas



(WP1 and WP7 support technical processes)

# CONTRACT Architecture



# CONTRACT Architecture



## Roles Supporting Contract Establishment

---

- The following roles can support the process of establishing a contract:
  - **Contract Validator** ensures the creation of legally-compliant contract instances
  - **Notary** is a trusted party that stores contract instances after the contract has been agreed upon, checked for validity and signed by both parties. Such contract instances may be later used as evidence of agreement in the contract monitoring and enforcement activities.
  - **Contract Repository** provides storage and access to standard contract forms. It can be used by parties to the contract who use pre-defined contract forms to produce individual contract instances or by contract drafters who are defining building blocks for contracts. There may be also a need for a specialised contract templates editor tools.

# CONTRACT Architecture



## Roles Supporting Contract Execution

---

- The following roles can support contract enforcement and performance monitoring during the performance of a contract.
  - **Contract Monitor** enables monitoring of the activities of parties, measuring their performance if needed and recording the relevant events.
  - **Notifier** implements various notifications mechanisms needed to send warning messages to indicate a pending contract-significant event, including possible non-compliance event that may be detected.
  - **Contract Enforcer** applies enforcing actions directly to the parties to ensure that some specific behaviour conforms to the contract
  - **Contract Lifecycle manager** can be queried against the status of a contract

# CONTRACT



## Contribution to DE

---

- Service Level Contracting Framework
  - Open specification for the DE community
- Service-Level Contracting Language Specification
  - Open specification for the DE community
- Service-Level Contracting Architecture, Tools and Plugins
  - Open specification of standard interfaces for developers
  - A complete code implementation will be available in OS
  - Plug-ins for open source development environments (e.g. Eclipse)
- Service-Level Verification and Monitoring
  - A complete code implementation will be available in OS

# CONTRACT Results



## Timeplan of public releases next year

---

- **June 07**
  - State of the Art
  - Use Case Outlines and Requirements
- **Sep 07**
  - Contracting Language Syntax and Semantics Specifications
  - Straw man Theoretical Framework Mapping to Web Services – Version I
- **Feb 08**
  - Revised Architecture Complete – Version II
  - Draft contracting language and algorithms specification II
  - WS framework for contract based computing – Version II
  - Alpha versions of core component prototypes
  - Alpha/Beta versions of contract verification mechanisms/tools

[ist-contract.org](http://ist-contract.org)