A FRAMEWORK FOR ANALYZING POLICY COMPLIANCE IN SOA SYSTEMS

Diploma Thesis of Konstantinos Angelopoulos
Advisor: Kostas Kontogiannis

National Technical University of Athens – Department Electrical and Computer Engineering

Problem Description

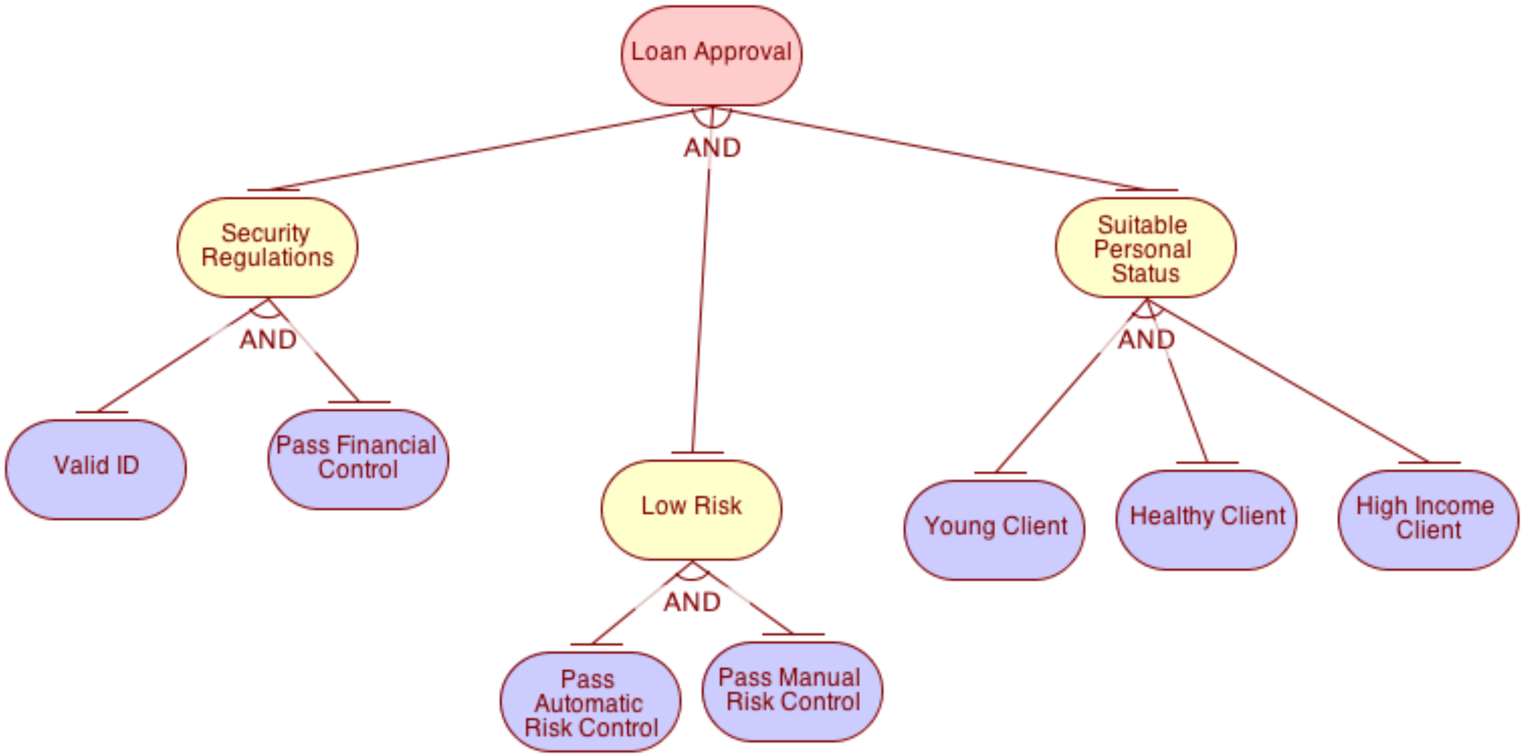
- IT systems that implement business processes should take into account the policies dictated by the stakeholders.
- The execution of the business processes should be compliant with the defined policies but also with the law and regulation restrictions that define the environment.
- Policies, laws and regulations are rapidly changed and may be in conflict sometimes, as the same business process execution runs in more than one environments. **Keyword: <u>Level of Compliace</u>**
- Manual compliance audit is expensive and sometimes requires the business processes to be stopped or paused (more expensive).
- SOA systems that implement business process are loosely coupled and this may lead to lack of information for a completely successful audit. **Keyword: <u>Uncertainty</u>**

Proposed Solution

- Model policies and restrictions using Goal-Models.
- Monitor the system's function and reclaim logged data for providing feedback to adjust the compliance control.
- Automate the compliance audit by using machine learning training to exploit the experience of past cases and reasoning techniques to diagnose compliance violations.
- Use probabilistic methods combined with First Order Logic (FOL) to face the uncertainty.

Modeling Policies and Regulations

A Model for a Loan Approval Business Process:



Handling Uncertainty

Markov Logic Networks (MLN) :

- Combine probabilistic graphic models with first order logic.
- Relax the hard constraint assumption on satisfying a formula.
- A possible world not satisfying a specific formula will simply be less likely.
- The more formulas a world satisfies, the more likely it is.
- Each formula can have a weight indicating how strong a constraint it should be for possible worlds.

$$P(\text{world}) \propto \exp\left(\sum \text{weights of formulas it satisfies}\right)$$

Markov Logic: Definition

- A Markov Logic Network (MLN) is a set of pairs (F, w) where:
  - F is a formula in first-order logic
  - w is a real number

- Together with a set of constants, it defines a Markov network with:
  - one node for each grounding of each predicate in the MLN
  - one feature for each grounding of each formula F in the MLN, with the corresponding weight w

Example: Friends & Smokers

Smoking causes cancer.

Friends have similar smoking habits.

Example: Friends & Smokers

$$\forall x \; Smokes(x) \Rightarrow Cancer(x)$$
$$\forall x, y \; Friends(x, y) \Rightarrow \big(Smokes(x) \Leftrightarrow Smokes(y)\big)$$

Example: Friends & Smokers

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

Example: Friends & Smokers

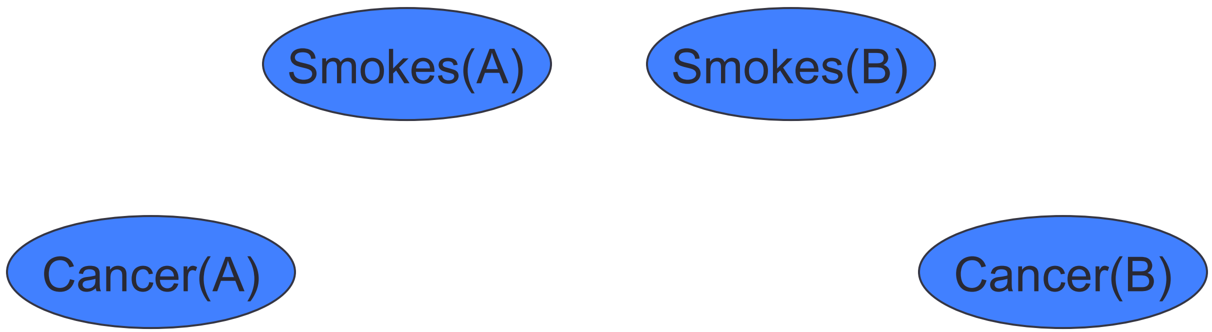
1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

Two constants: **Anna** (A) and **Bob** (B)

Example: Friends & Smokers

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

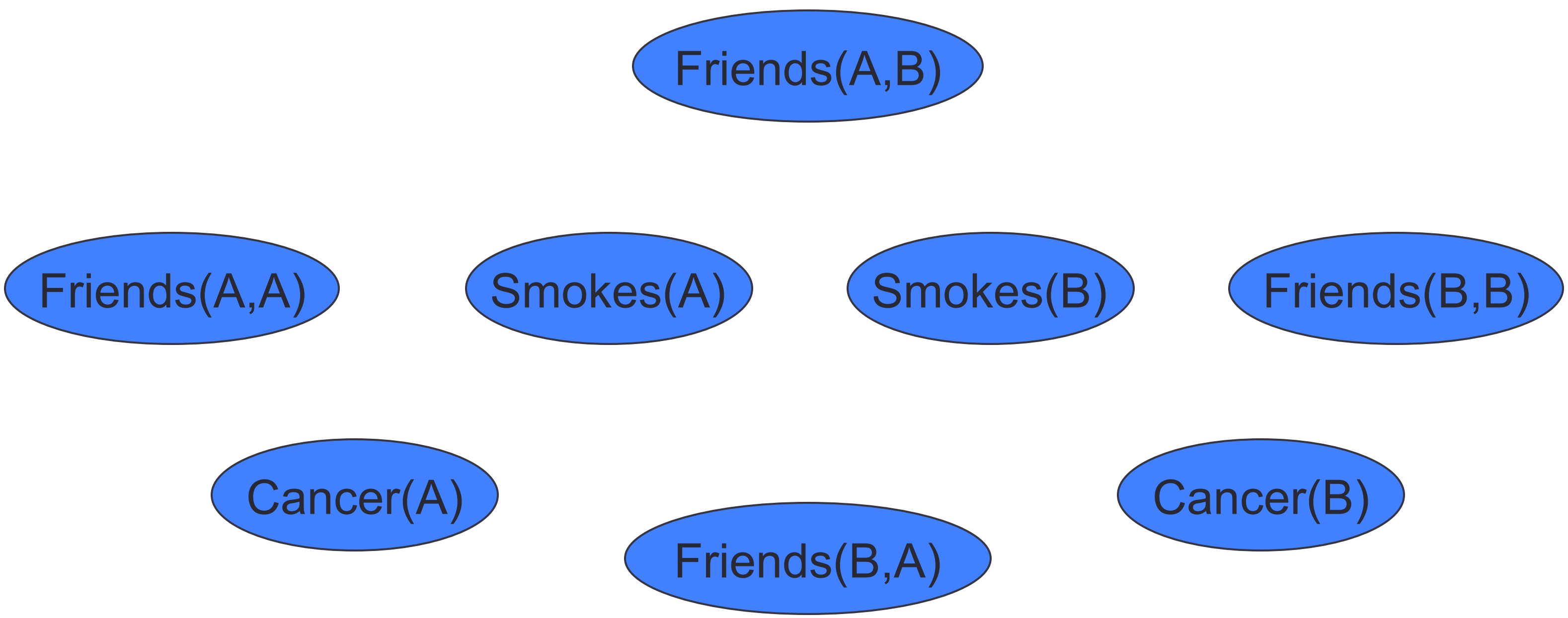
Two constants: **Anna** (A) and **Bob** (B)



Example: Friends & Smokers

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

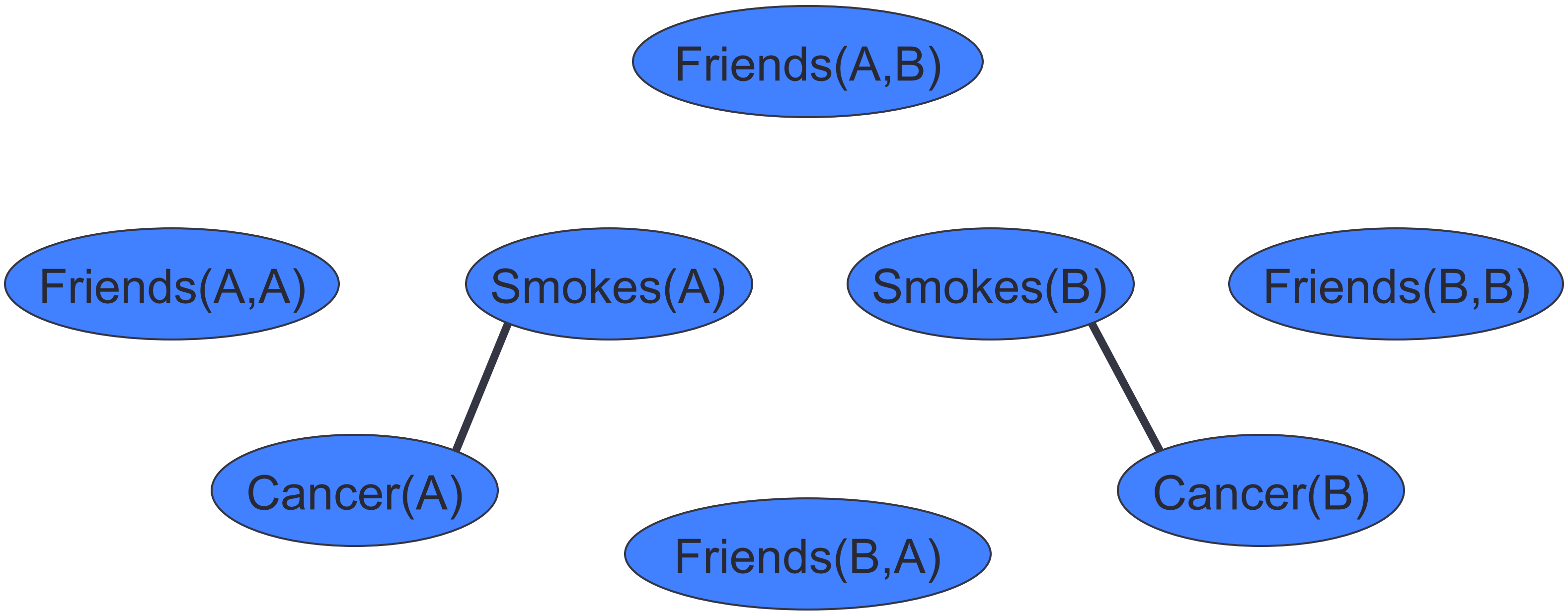
Two constants: **Anna** (A) and **Bob** (B)



Example: Friends & Smokers

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow \big(Smokes(x) \Leftrightarrow Smokes(y)\big)$

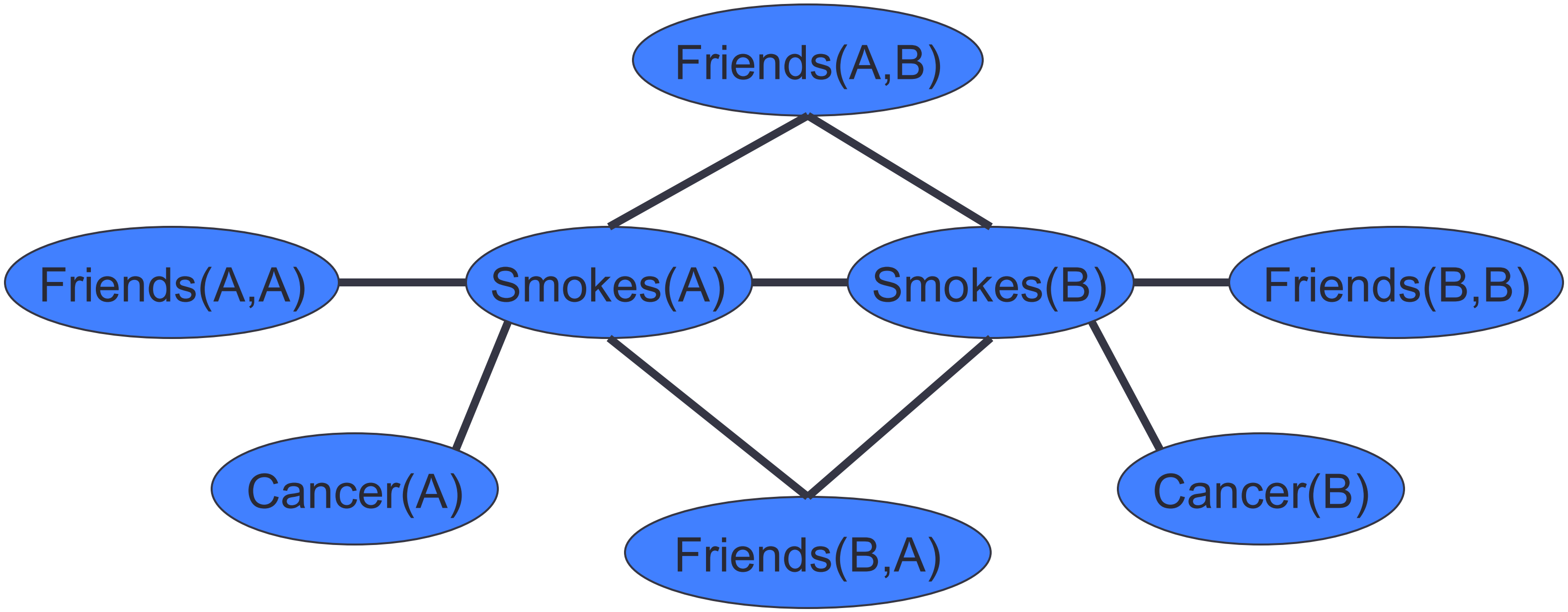
Two constants: **Anna** (A) and **Bob** (B)



Example: Friends & Smokers

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

Two constants: **Anna** (A) and **Bob** (B)



Markov Logic Networks

- MLN is **template** for ground Markov nets

- Probability of a world x:

$$P(x) = \frac{1}{Z} \exp\left(\sum_i \boxed{w_i} \boxed{n_i(x)} \right)$$

Weight of formula *i*

No. of true groundings of formula *i* in *x*

Markov Logic Networks

- How do we define the weights for each formula?
  - **Manually** (not possible for complex rules)
  - **Learning** (using **training datasets**)

- How do calculate the probability of a possible world?
  - **Probabilistic inference** (based on the previous equation, using **evidence**)

- The inference and learning operations for Markov Logic Networks are supported by many open source tools and implement different types of algorithms according to the desired accuracy or performance.

Putting It All Together

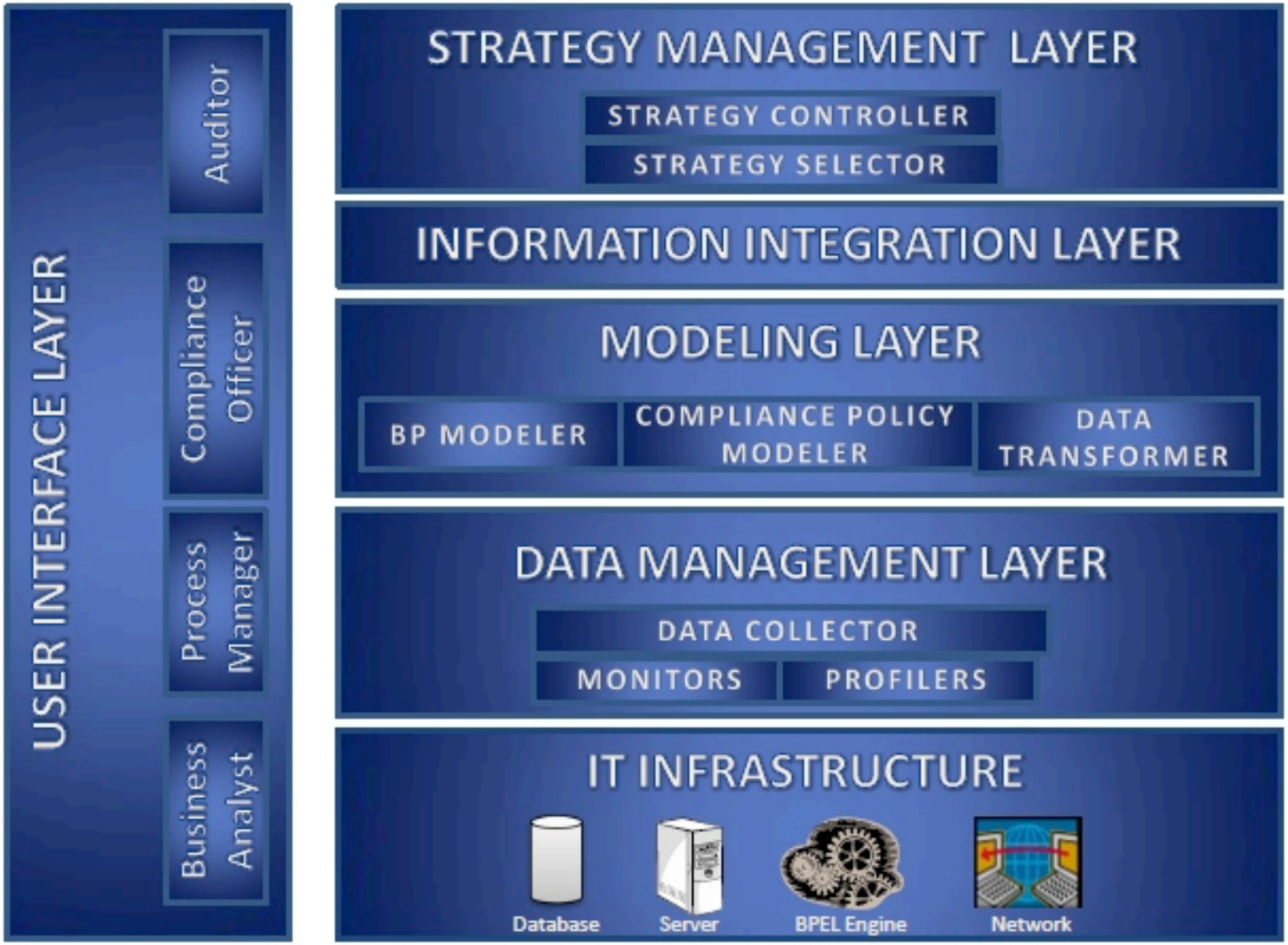
Step 1 : Design the Goal Models and relate the nodes with first order logic predicates or formulas.

Step 2 : Convert the Goal Models into Markov Logic Networks and define the formula weights by using training datasets.

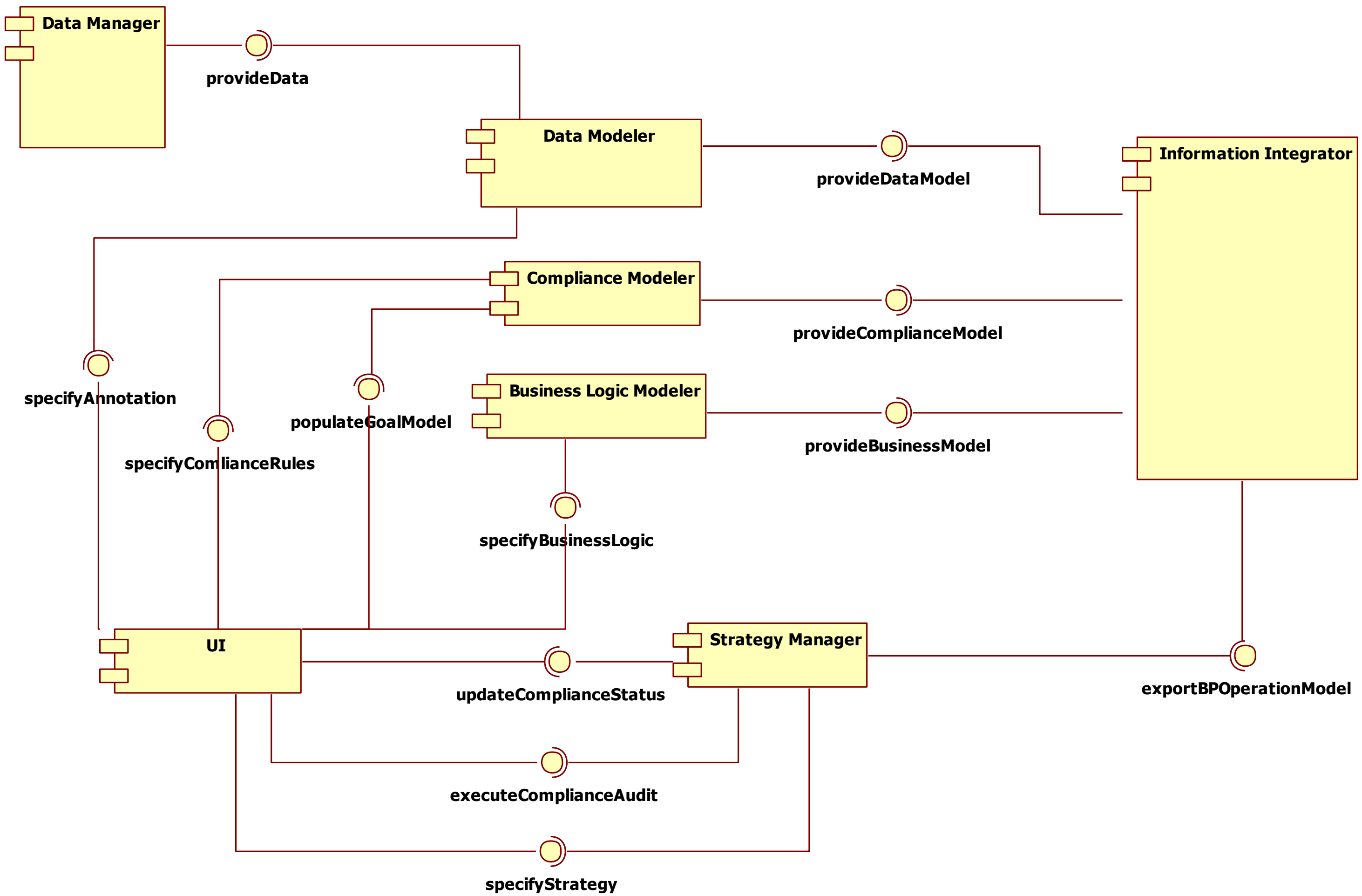
Step 3 : Monitor the execution of the business process, log the events and convert them to grounded predicates, creating an evidence knowledge base.

Step 4 : Apply probabilistic inference on the produced Markov Logic Networks, based on the evidence knowledge base, to calculate the probability of the system to be compliant with the specific policy.

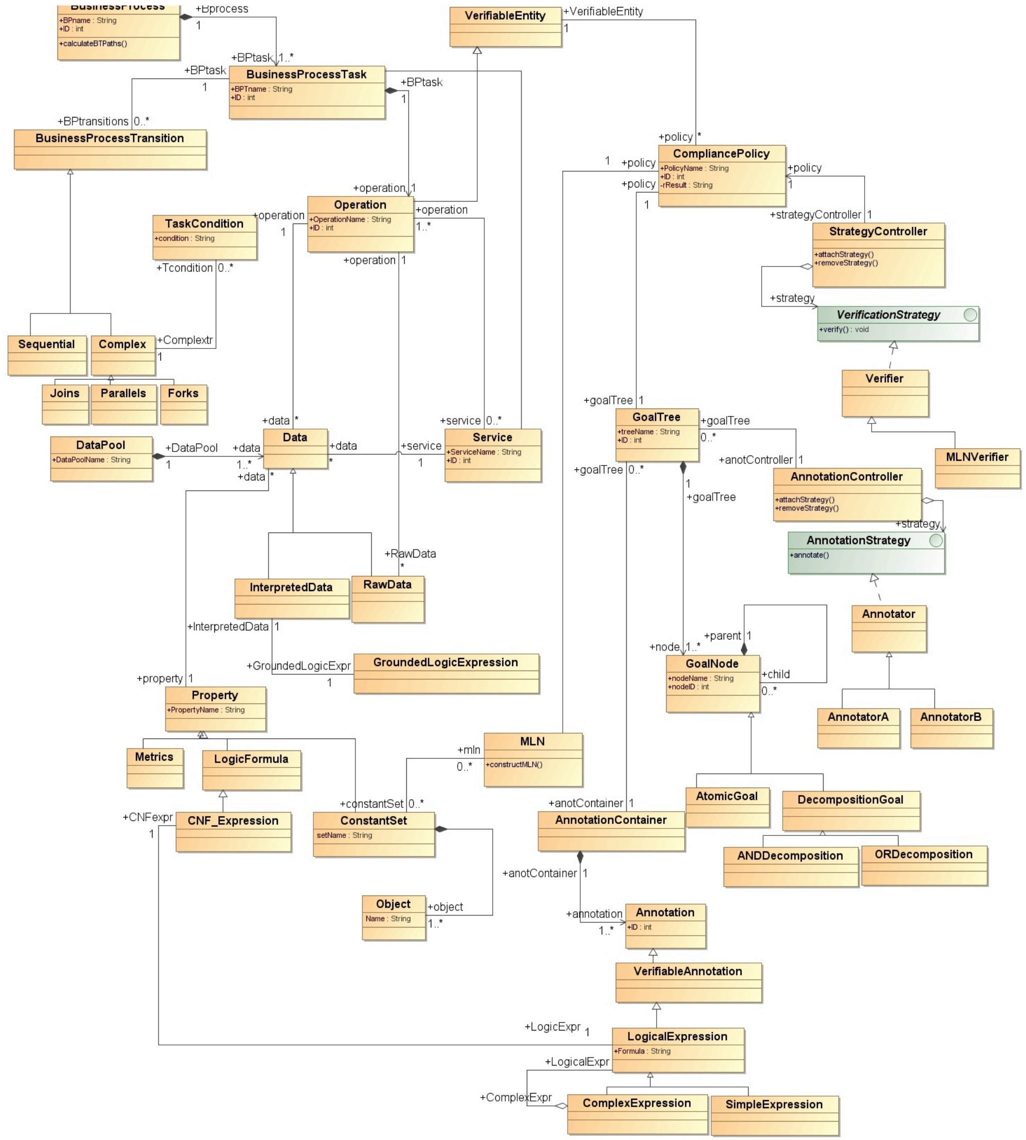
Framework's Architecture



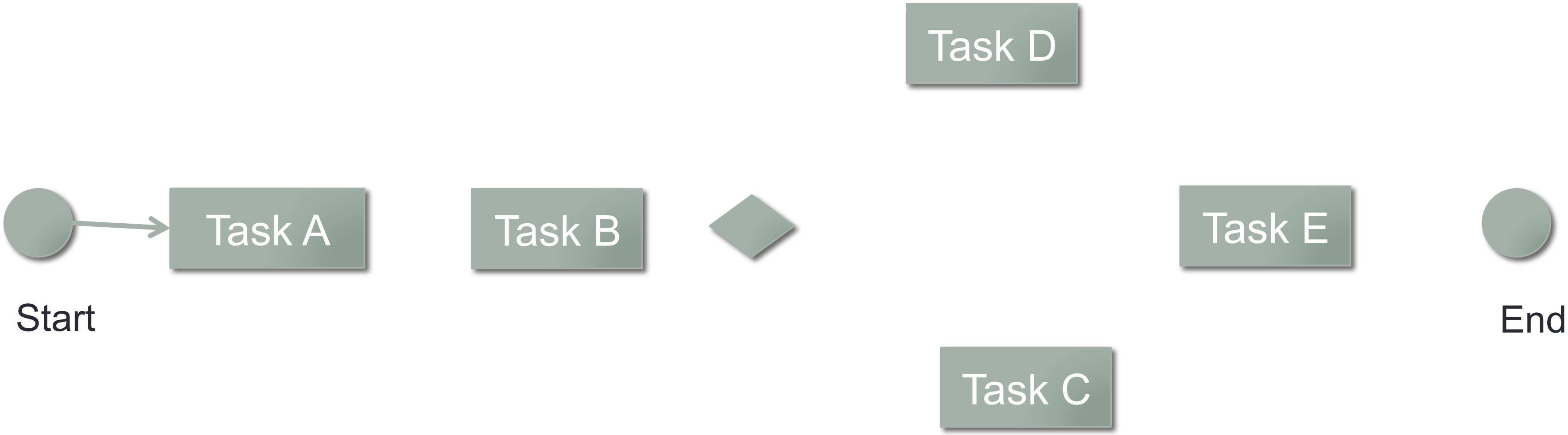
Framework's Architecture



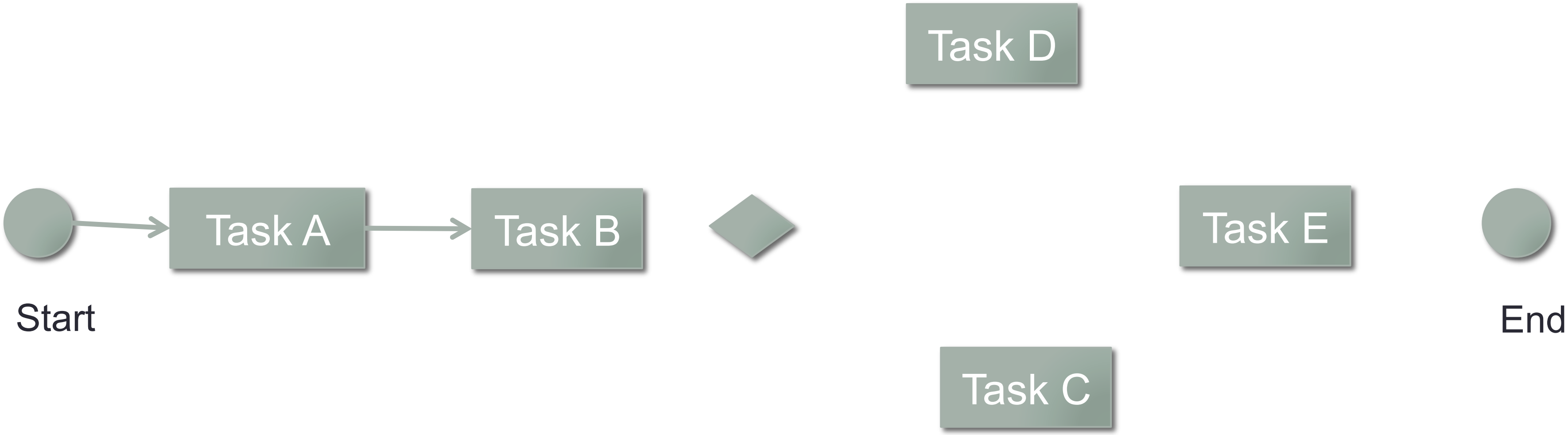
Domain Model



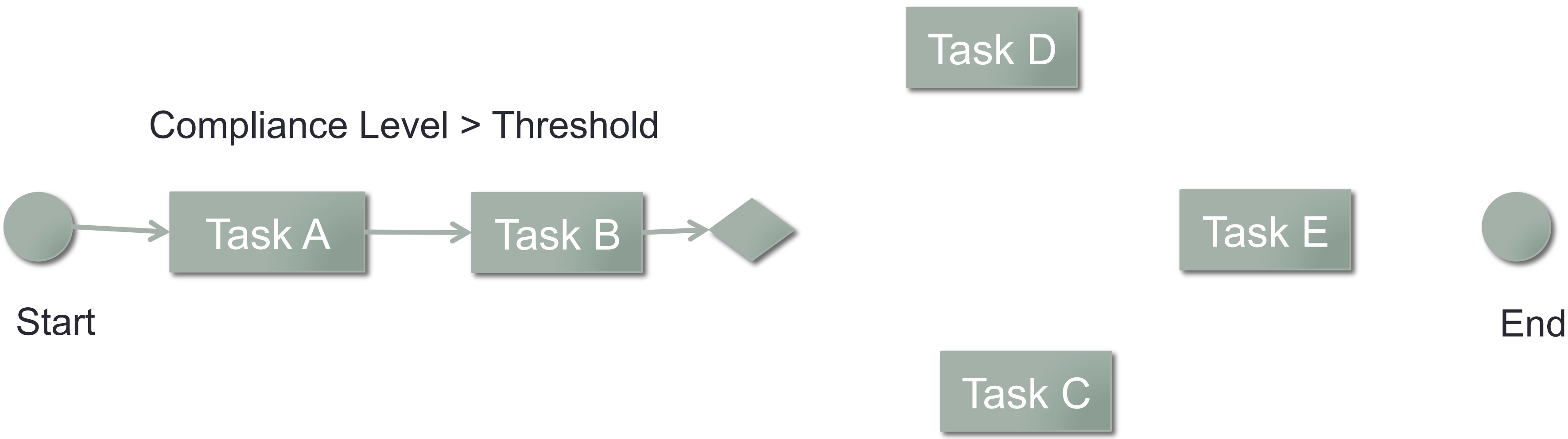
General Execution Idea



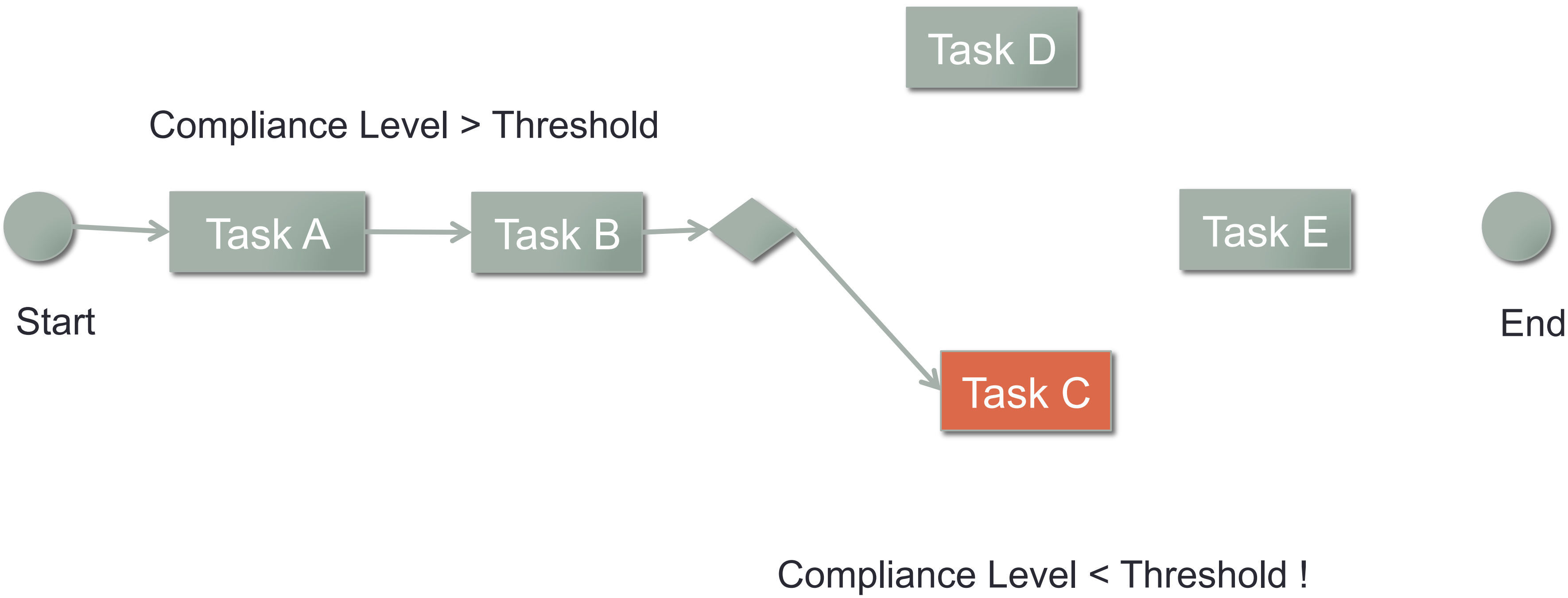
General Execution Idea



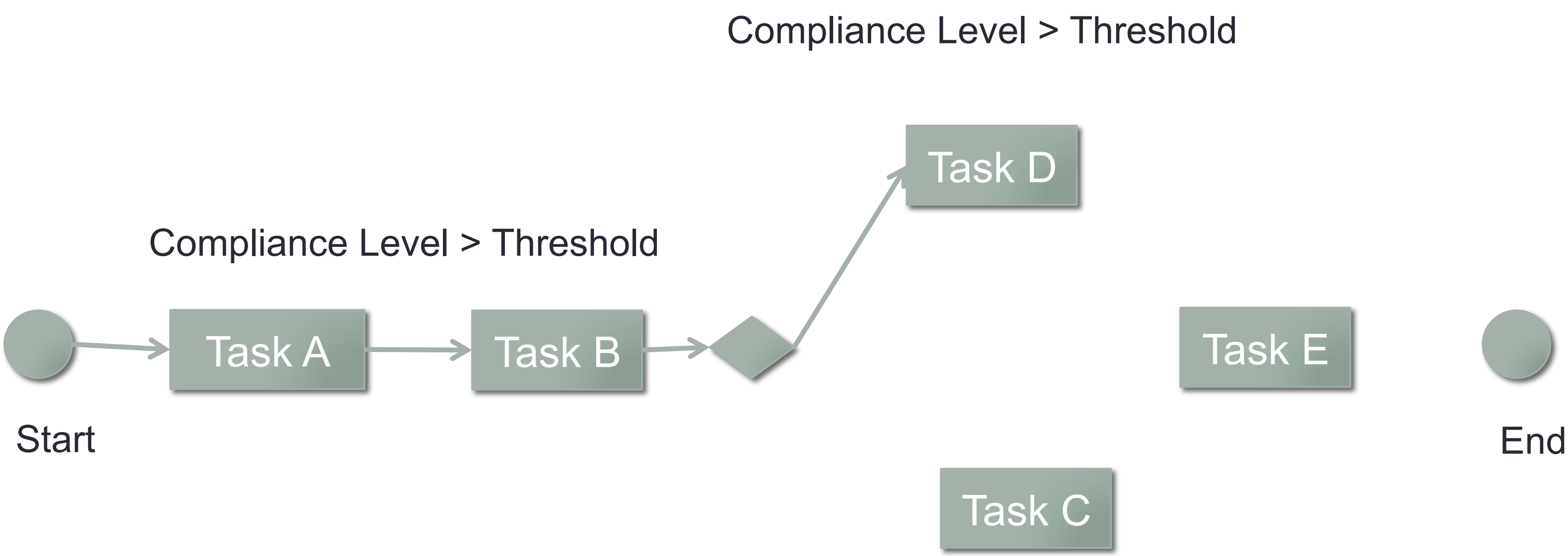
General Execution Idea



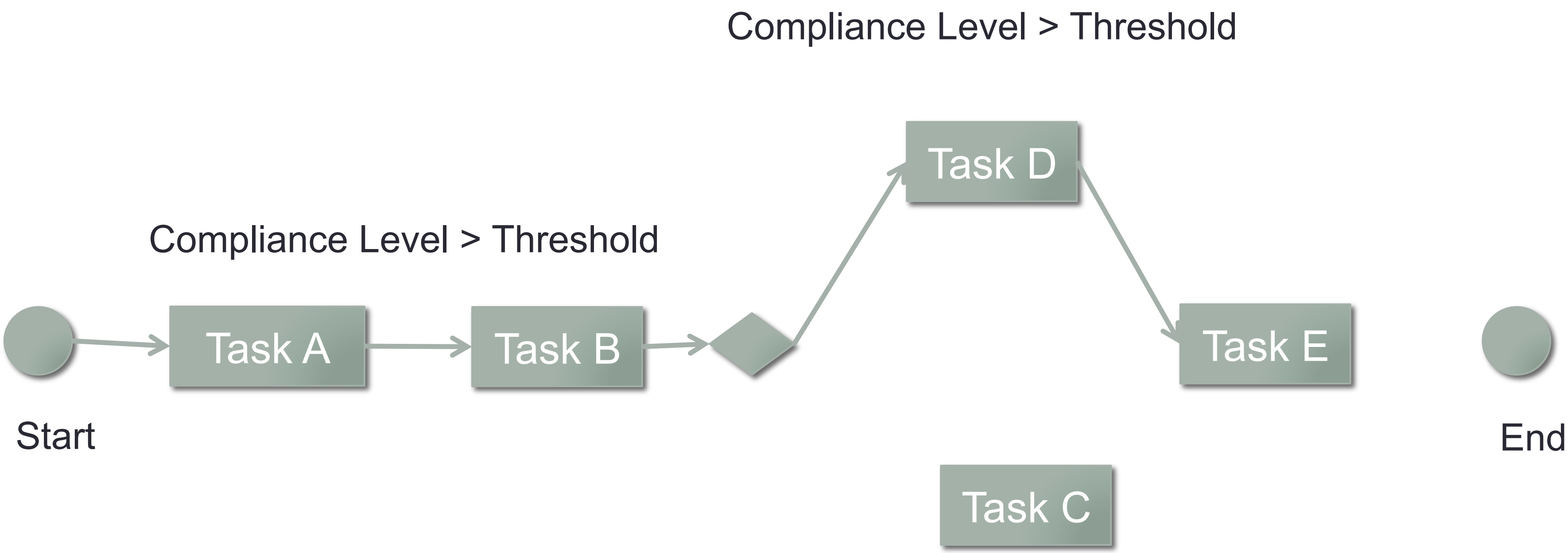
General Execution Idea



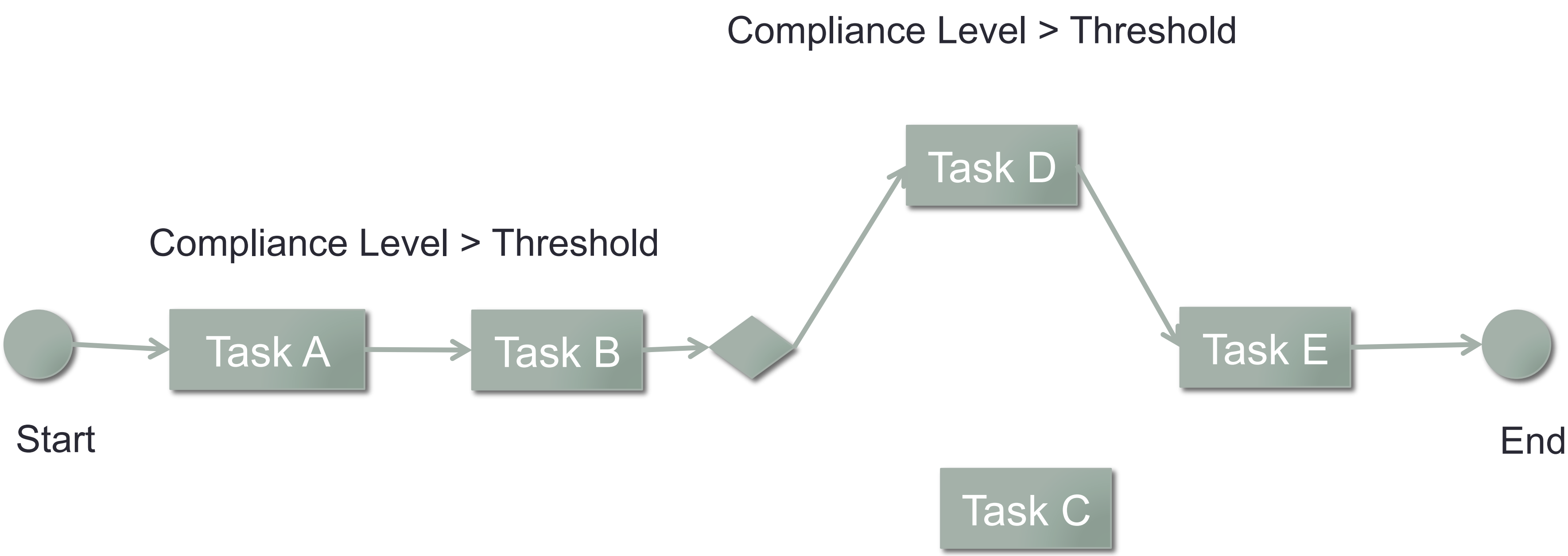
General Execution Idea



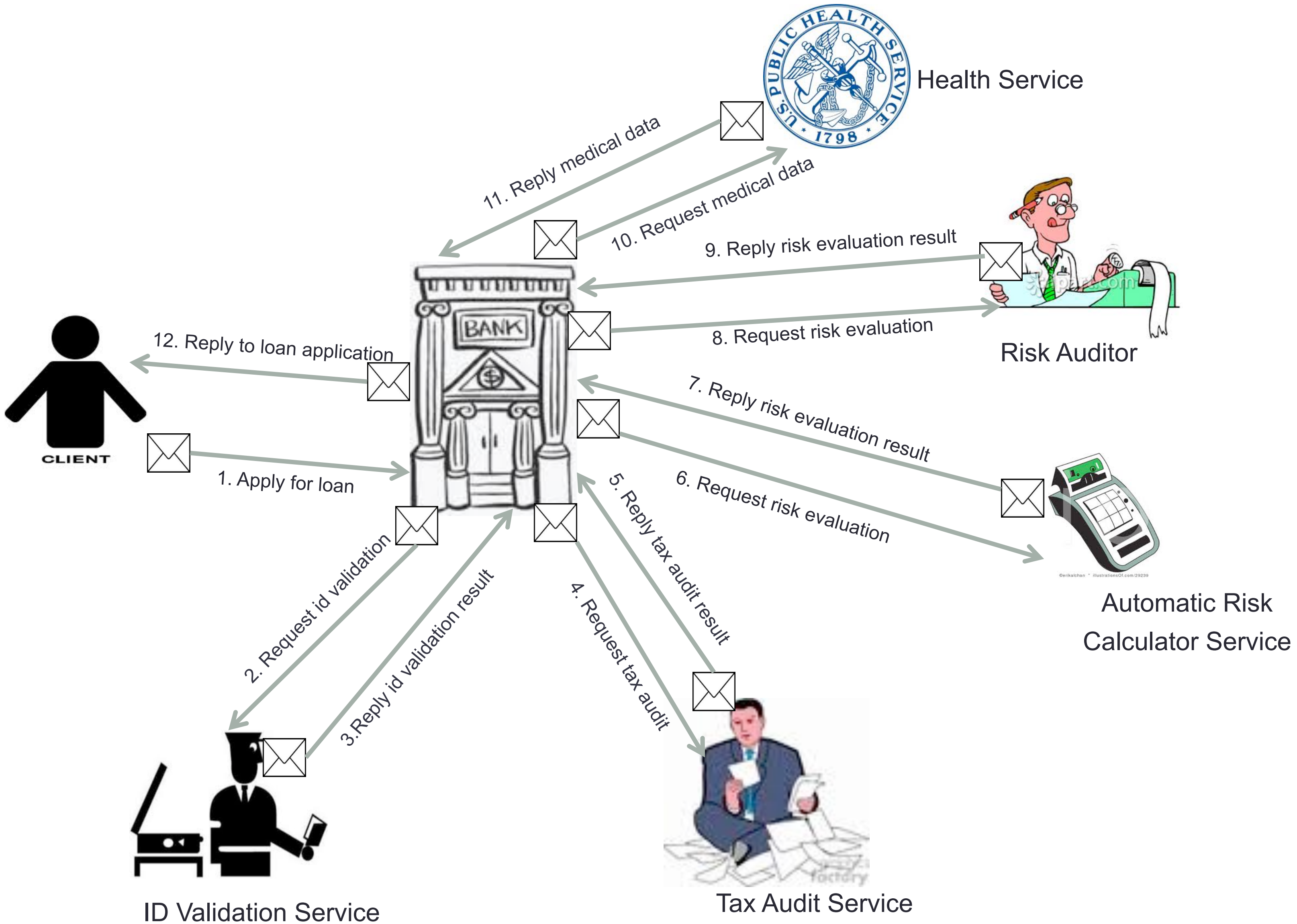
General Execution Idea



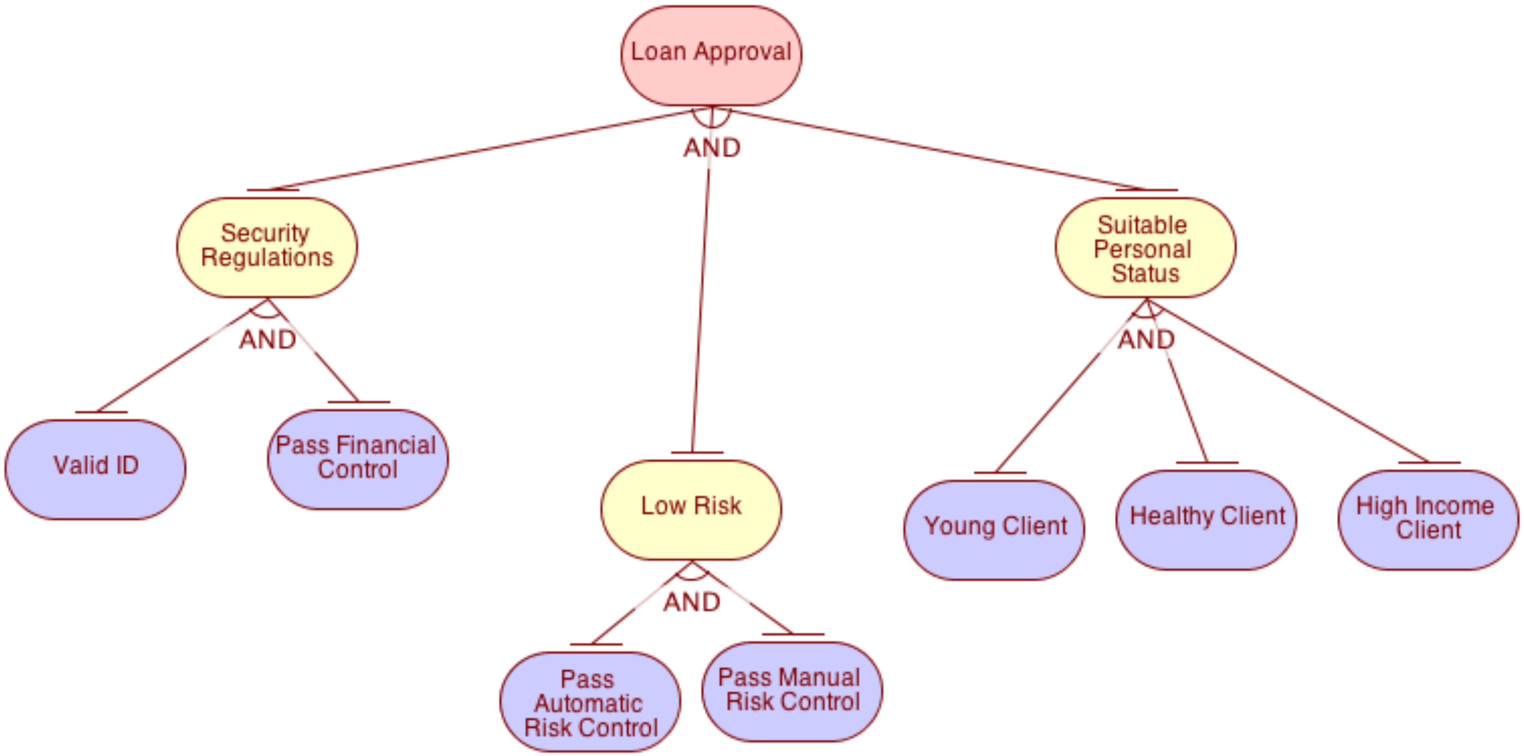
General Execution Idea



Use Case: The Business Process



Use Case: Goal Model



Use Case: Goal Conversion

Converting Goals to predicates and formulas

G: Loan Approval	➔ LoanApproval(x)
G: Security Regulations	➔ SecurityRegulations(x)
G: Low Risk	➔ LowRisk(x)
G: Suitable Personal Status	➔ SuitableStatus(x)
G: Valid ID	➔ ValidatedID(x)
G: Pass Financial Control	➔ PassFinancialControl(x)
G: Pass Automatic Risk Control	➔ AutoCalculatedRisk(x,y)∧lessThan(y, 4) => PassAutoRiskControl(x).
G: Pass Manual Risk Control	➔ ManualyCalculatedRisk(x,z)∧lessThan(z, 4)=>PassManualRiskControl(x)
G: Young Client	➔ Age(x,w)∧lessThan(w, 47)=> Young(x)
G: Healthy Client	➔!LifeThreateningDisease(x) ∧ !InheritedDisease(x)∧ !RecentlyHospitalised(x) => Healthy(x)
G: High Income Client	➔Income(x,k) ∧ greaterThan(k,2)=> HighIncome(x)

Use Case: Generated MLN

```
risk = {0,…,5}
age = {18,…,100}
incomeCategory = {1,…,5}

LoanApproval(bp)
SecurityRegulations(bp)
LowRisk(bp)
SuitableStatus(bp)
ValidatedID(bp)
PassFinancialControl(bp)
PassAutoRiskControl(bp)
PassManualRiskControl(bp)
AutoCalculatedRisk(bp,risk)
ManualyCalculatedRisk(bp,risk)
Age(bp,age)∧lessThan(age, 47)=> Young(bp)
!LifeThreateningDisease(bp) ∧ !InheritedDisease(bp)∧ !RecentlyHospitalised(bp) => Healthy(bp)
Income(bp,incomaCategory) ∧ greaterThan(incomeCategory,2)=> HighIncome(bp)

SecurityRegulations(x) ∧ LowRisk(x) ∧ SuitableStatus(x) => LoanApproval(x).
ValidateID(x) ∧ PassFinancialControl(x)=>SecurityRegulations(x).
PassAutoRiskControl(x) ∧ PassManualRiskControl(x) => LowRisk(x).
Young(x) ∧ Healthy(x) ∧ HighIncome(x) => SuitableStatus(x).
AutoCalculatedRisk(x,y) ∧lessThan(y, 4)=>PassAutoRiskControl(x).
ManualyCalculatedRisk(x,z) ∧lessThan(z, 4)=>PassManualRiskControl(x).


Age(x,w)∧lessThan(w, 47)=> Young(x).
!LifeThreateningDisease(x) ∧ !InheritedDisease(x)∧ !RecentlyHospitalised(x) => Healthy(x).
Income(x,k) ∧ greaterThan(k,2)=> HighIncome(x).
```

Use Case: Execution 1

Evidence Knowledge Base

Age(BP1,32)
!LifeThreateningDisease(BP1)
!InheritedDisease(BP1)
!RecentlyHospitalised(BP1)
Income(BP1,5)
AutoCalculatedRisk(BP1,5)
ManualyCalculatedRisk(BP1,1)
ValidateID(BP1)
PassFinancialControl(BP1)

Probabilistic Inference Result

P(LoanApproval(BP1)|Evidence) =0.225027

Use Case: Execution 2

Evidence Knowledge Base

Age(BP1,32)
!LifeThreateningDisease(BP1)
!InheritedDisease(BP1)
!RecentlyHospitalised(BP1)
Income(BP1,1)
AutoCalculatedRisk(BP1,5)
ManualyCalculatedRisk(BP1,1)
ValidateID(BP1)
PassFinancialControl(BP1)

Probabilistic Inference Result

P(LoanApproval(BP1)|Evidence) =0.00184982

Use Case: Execution 3

Evidence Knowledge Base

Age(BP1,32)
!LifeThreateningDisease(BP1)
…Missing Information….
Income(BP1,1)
AutoCalculatedRisk(BP1,5)
ManualyCalculatedRisk(BP1,4)
ValidateID(BP1)
PassFinancialControl(BP1)

Probabilistic Inference Result

P(LoanApproval(BP1)|Evidence) =0.79807

Use Case: Execution 4

Evidence Knowledge Base

Age(BP1,32)
!LifeThreateningDisease(BP1)
!InheritedDisease(BP1)
!RecentlyHospitalised(BP1)
Income(BP1,5)
AutoCalculatedRisk(BP1,3)
ManualyCalculatedRisk(BP1,1)
ValidateID(BP1)
PassFinancialControl(BP1)

Probabilistic Inference Result

P(LoanApproval(BP1)|Evidence) =) 0.99995

Questions?