Aligning Service-Oriented Architectures with Security Requirements

Mattia Salnitri¹, Fabiano Dalpiaz², Paolo Giorgini¹

¹University of Trento, Italy ²University of Toronto, Canada



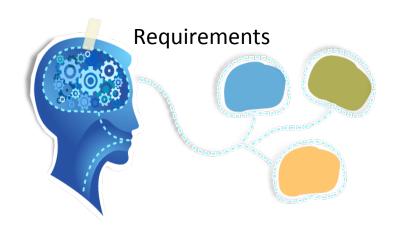


Outline

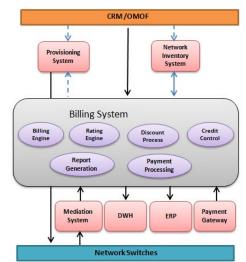
- Change
 - Requirements
 - Service-oriented Architectures
- Modelling SoA and Security Requirements
- Our proposal
- Example
- Future works and conclusions

Change



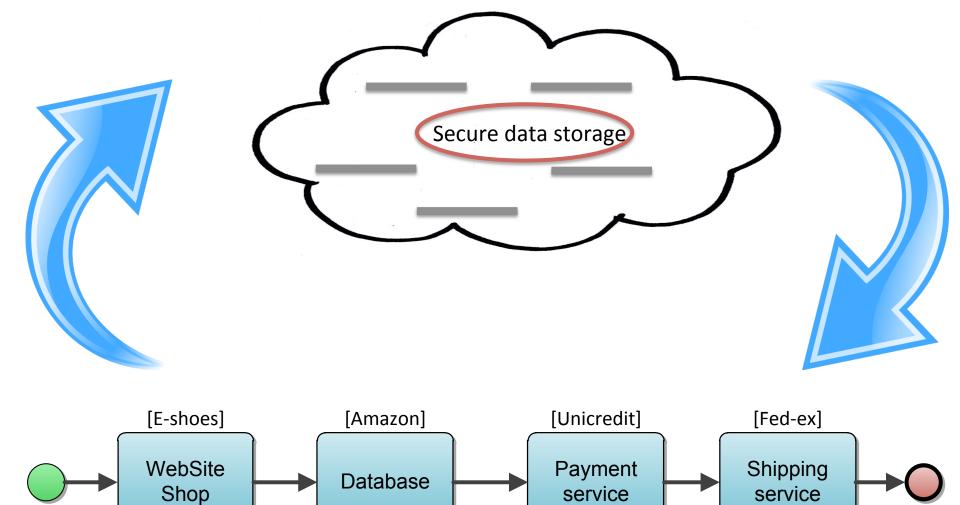


Software

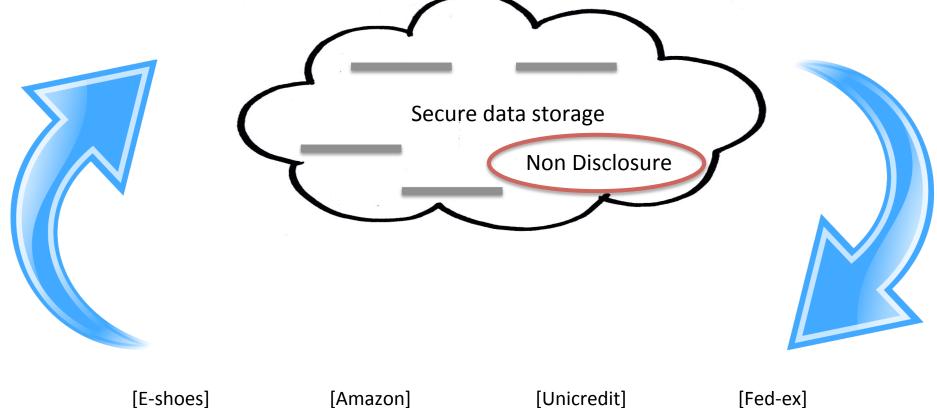


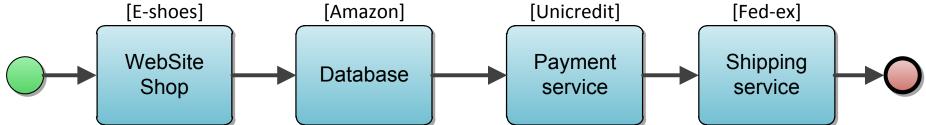


Motivating Example



Motivating Example





Context: the Aniketos project

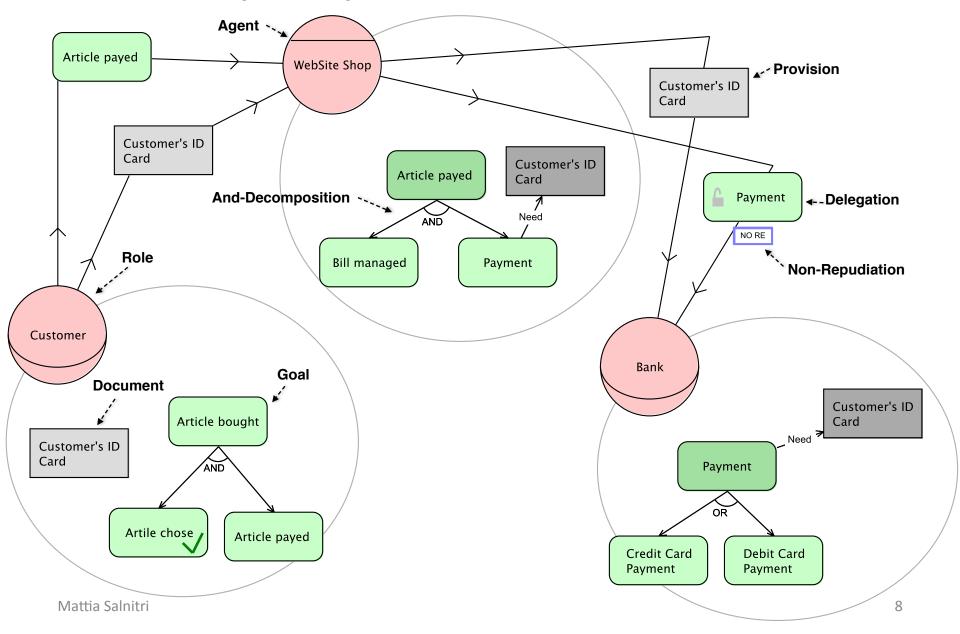
- Secure service compositions in a constantly changing environment
 - Design-time:
 - specification of the security requirements
 - Run-time:
 - creation of a secure service composition
 - alignment of requirements with evolved service compositions
- For more information : http://www.aniketos.eu/



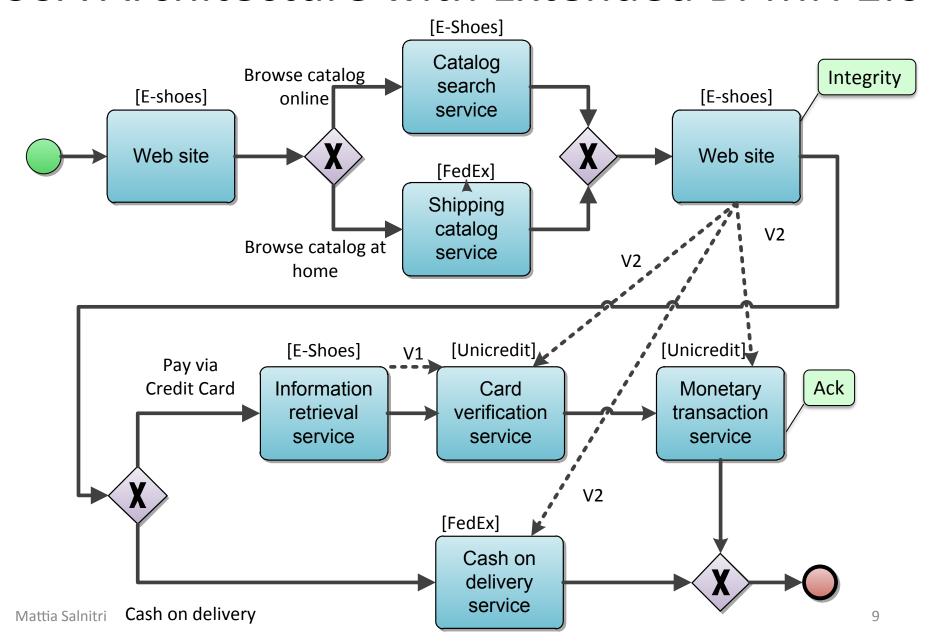


Modelling language overview Municipal approval contract Seller Municipality notified Tangible By NO RE AND-Decomposition Tangible By notified Official contract Sale information Seller Land sold Building PartOf PartOf approval Contract draft Contract draft Int parties Draft prepared Land details Price Official contract Approva AND Government notified Building approval Produce Buyer notified Municipality Contract draft Official contract checked updated Need Contract draft Building Permitted Operations (Use, | > U M P D Modify, Produce, Distribute Municipality U M P D Seller eGov applicatio U M P D V2 - [Athens REA] Land details Municipal approval Sale information V1 [Athens Munic.] [Athens REA] [Athens REA] Government notified Create Insert Examine Add duty approval contract draft stamp draft [Athens REA] [Greek ministry] [eGov] Requirements Send Check Review copy to contract copy ministry [Storage] [eGov] Archive Scan contract contract **Architecture** Mattia Salnitri copy

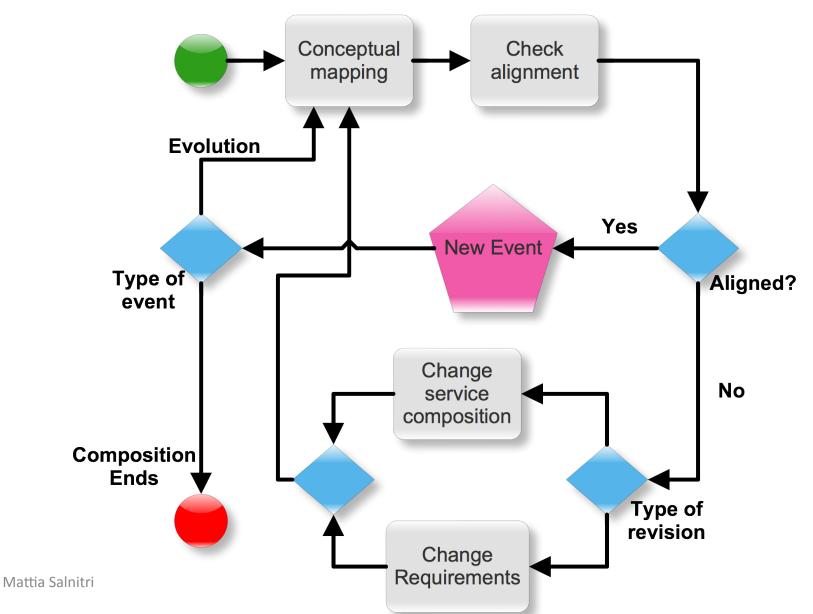
Security requirements with STS-ml



SoA Architecture with Extended BPMN 2.0



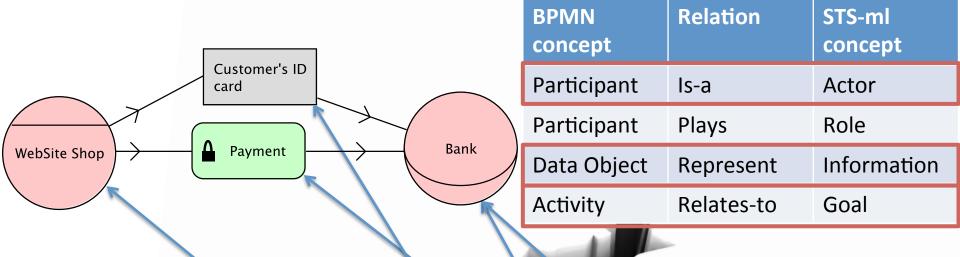
Alignment methodology



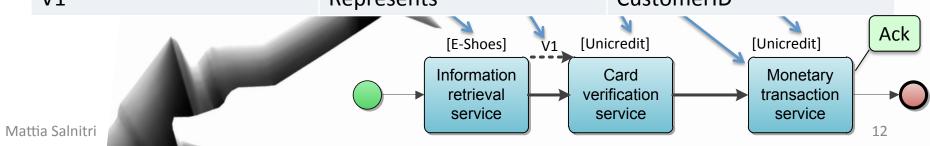
Conceptual mapping

BPMN 2.0 Graphical element	BPMN concept	Relation	STS-ml concept	STS-ml Graphiacl elements
[Participant]	Participant	ls-a	Actor	Agent
[Participant]	Participant	Plays	Role	Role
_ [Data Object]_>	Data Object	Represent	Information	Document
Service	Activity	Relates-to	Goal	Goal

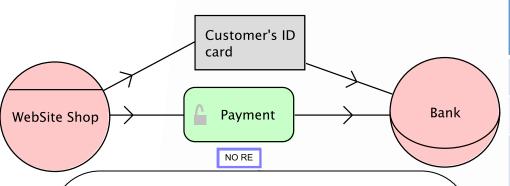
Conceptual mapping example



Business Process Element	Relation	STS-ml Elements
Unicredit	Is-a	Bank
E-shoes	Is-a	WebSite Shop
Money transaction	Relates-to	Payment
V1	Represents	CustomerID



Alignment check example



Business Process Element	Relation	STS-ml Elements
Unicredit	Is-a	Bank
E-shoes	Is-a	WebSite Shop
Money transaction	Relates-to	Payment
V1	Represents	CustomerID

/	Algorithm 5 Non-repudiation verification				
	Ver	Verifynr(actnr, perf, actCurr, found, visited)			
	1	switch TypeOf($actCurr$)			
	2	case ack:			
	3	$if (actNR \in AckFor(actCurr) \land Perf(actCurr) = perf)$			
	4	then return true			
	5	case end:			
	6	return not found			
	7	case default :			
	8	if (actCurr = actNR)			
	9	$\mathbf{then}\ found \leftarrow \mathtt{true}$			
	10	$next \leftarrow \text{NextActivities}(actCurr) \setminus visited$			
	11	if $next = \emptyset$ then return true			
	12	switch TypeOf(NextElement($actCurr$))			
	13	case gway-excl:			
	14	return $\bigwedge_{a \in next} VerifyNR(actNR, perf, a, found, visited \cup actCurr)$			
	15	case gway-incl:			
	16	return $\bigvee_{a \in next} VerifyNR(actNR, a, perf, found, visited \cup actCurr)$			
	17	case activity:			
	18	VerifyNR($actNR$, $perf$, GetFirst($next$), $found$, $visited \cup actCurr$)			
1					

Mattia Salnitri

[E-Shoes] V1 [Unicredit] [Unicredit]

Information retrieval service | Card verification service | Card verificatio

Non repudiation

Algorithm 5 Non-repudiation verification

```
VerifyNR(actNR, perf, actCurr, found, visited)
     switch TypeOf(actCurr)
       case ack:
           if (actNR \in AckFor(actCurr) \land Perf(actCurr) = perf)
               then return true
       case end:
           return not found
       case default:
           if (actCurr = actNR)
  8
 9
              then found \leftarrow \texttt{true}
     next \leftarrow \text{NextActivities}(actCurr) \setminus visited
10
     if next = \emptyset then return true
12
     switch TypeOf(NextElement(actCurr))
13
       case gway-excl:
           return \bigwedge_{a \in next} VerifyNR(actNR, perf, a, found, visited \cup actCurr)
14
15
       case gway-incl:
           return \bigvee_{a \in next} VerifyNR(actNR, a, perf, found, visited \cup actCurr)
16
17
       case activity:
           VerifyNR(actNR, perf, Getfirst(next), found, visited \cup actCurr)
18
```

Search for an ack,
If needed

Recursively inspect the business process

Non distribution

Algorithm 3 Non-Disclosure Verification

```
VERIFYND(C(deb, cred, \top, non-discl(var)), BP, SRS, CM)
     actByDeb \leftarrow BP.ACTIVITIESBY(deb)
     actByCred \leftarrow BP.ActivitiesBy(cred)
     actUsingVar \leftarrow BP.ACTIVITIESUSING(var)
  3
     doc \leftarrow CM.Search(represents(var, *))
     if doc \neq null
  5
        then info \leftarrow SRS.SEARCH(tangible-by(*, doc))
 6
              for each i \in info
 8
              do own \leftarrow SRS.SEARCH(owns(*, i))
  9
                   actByOwner.Add(BP.ActivitiesBy(own))
     actByOthers \leftarrow actUsingVar \setminus actByDeb \setminus actByCred \setminus actByOwner
10
11
     for each a_i \in actByDeb
12
     do for each a_i \in actByOthers
13
         do if var \in output(a_i) \cap input(a_j)
14
                then return non-compliant
15
     return compliant
```

Retrieve all the message flows involved

Examine all the message flows retrieved

Alignment checking tool

- Security Requirements Compliance Module (SRCM)
 - Run-time/design-time module in Aniketos
 Framework
 - OSGi services, apache Karaf compliant
 - It currently supports various security requirements:
 - Non-repudiation
 - Non-distribution
 - Non-modification

Related Works

- Law compliance Ghanavati[2009]
 - Design time methodology
 - No security requirements
- Business process generation Desai [2004], Zeng[2002]
 - No evolution
- Auditing Business Process Compliance Ghose [2007]
 - No requirements
- Static Compliance-Checking Framework for Business Process Models Liu [2007]
 - Design time methodology

Future Work

- Extend the list of security requirements supported
- Validate our methodology and SRCM with three Aniketos industrial case studies
- Verification of security properties with symbolic execution techniques
- Alignment between systems

Mattia Salnitri

18

Conclusions

- Key feature
 - we propose run-time methodology to check the alignment of security requirements with service compositions
- Limitations
 - Scalability of semi-automated part
 - Support further security requirements

Thank you!

References

- **Ghanavati[2009]**: Sepideh Ghanavati, Daniel Amyot, and Liam Peyton. Compliance Analysis Based on a Goal-oriented Requirement Language Evaluation Methodology. In Proc. of RE'09, pages 133–142, 2009.
- **Ghose[2007]**: Aditya Ghose and George Koliadis. Auditing Business Process Compliance. In Proc. of ICSOC'07, pages 169–180, 2007.
- **Liu [2007]**: Y. Liu, S. Mü'ller, and K. Xu. A Static Compliance-Checking Framework for Business Process Models. IBM Systems Journal, 46(2):335–361, 2007.
- **Desai [2004]:** N. Desai, A.U. Mallya, A.K. Chopra and M.P. Singh. Processes = Protocols + Policies: A methodology for business process development. Un Proc. Of WWW'05, 2005.
- Zeng[2002]: L. Zeng, D. Flaxer, H. Chang and J. Jeng. PLM flow –
 Dynamic Business Process Composition and Execution by Rule
 Inference. In Proc. of TES'02, pages 141--150