



Some thoughts for future RTD in secure software systems and services

Workshop on

Software and Service Development, Security & Dependability

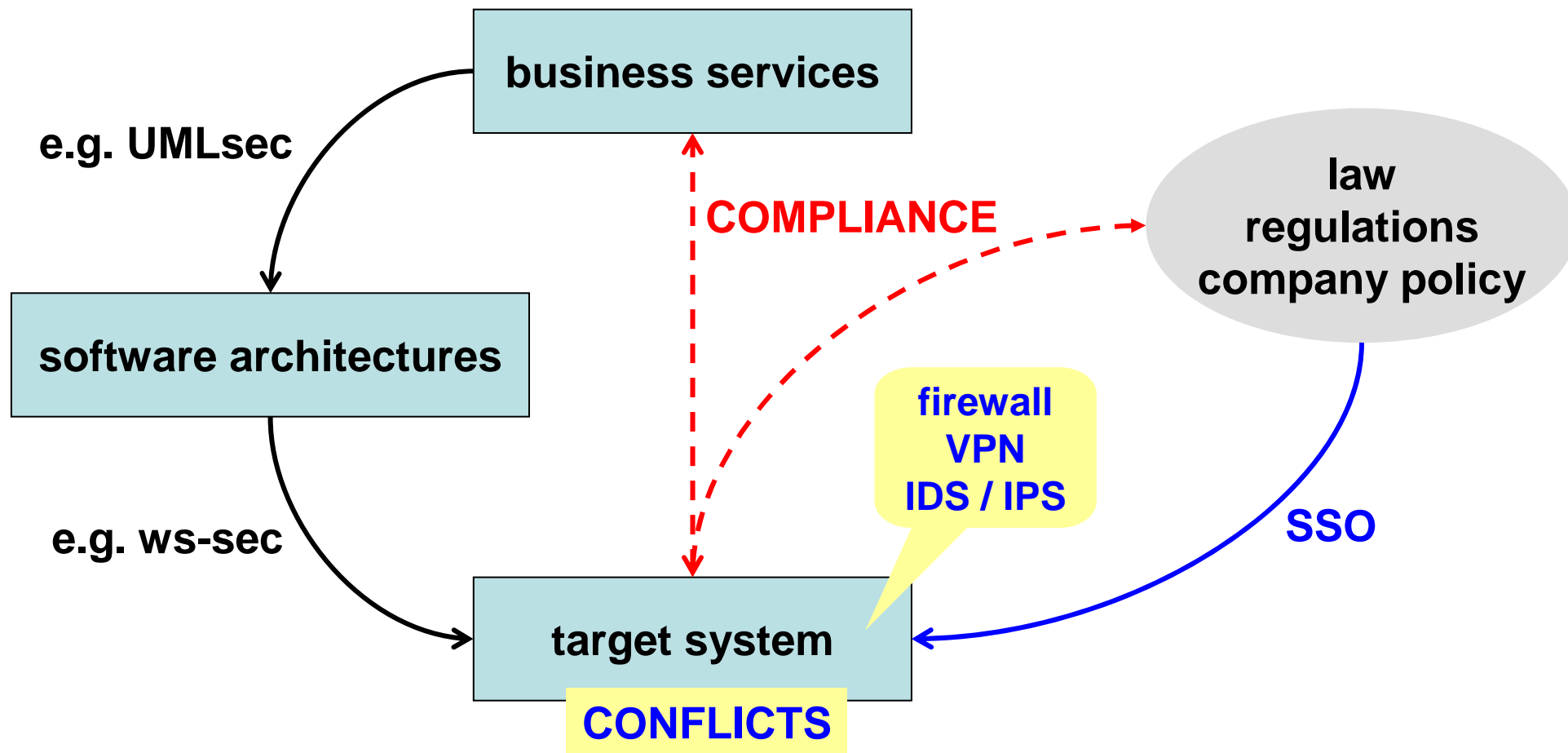
Antonio Lioy
Politecnico di Torino

Lioy@polito.it



Services, software and hardware

ESFORS Software and Service Development, Security & Dependability Workshop



Conflict example #1

ESFORS Software and Service Development, Security & Dependability Workshop

- ***my service is secure because it uses WS-security***
 - e.g. SOAP over HTTPS
 - e.g. XMLEncryption
- ... but the SSO says “any kind of encryption is forbidden on the Intranet because it prevents data inspection by the IDS”

Conflict example #2

ESFORS Software and Service Development, Security & Dependability Workshop

- ***this service is based on RPC and therefore it will use a random port in the range 1024-65535***
- ***please allow all these ports through the firewall***
- and the answer of the SSO is ...
 - YES = no security
 - NO = no service

Conflict example #3

ESFORS Software and Service Development, Security & Dependability Workshop

- ***this service runs on port 80/tcp***
- ***please allow all this port through the firewall***
- what's the problem here?
 - packet filter is happy ... but low security
 - application gateway wants to know application protocol (e.g. SOAP over HTTP)
 - semantic firewall and IDS wants to know authorized SOAP messages and users

Design methodologies and security

ESFORS Software and Service Development, Security & Dependability Workshop

- design methodologies must be security-aware
- wrt the underlying system, sw architects must know:
 - its security capabilities
 - its security constraints
- automate design (and config and mgmt) as much as possible
 - e.g. automatic fw design (min config ... max sec)

Automation

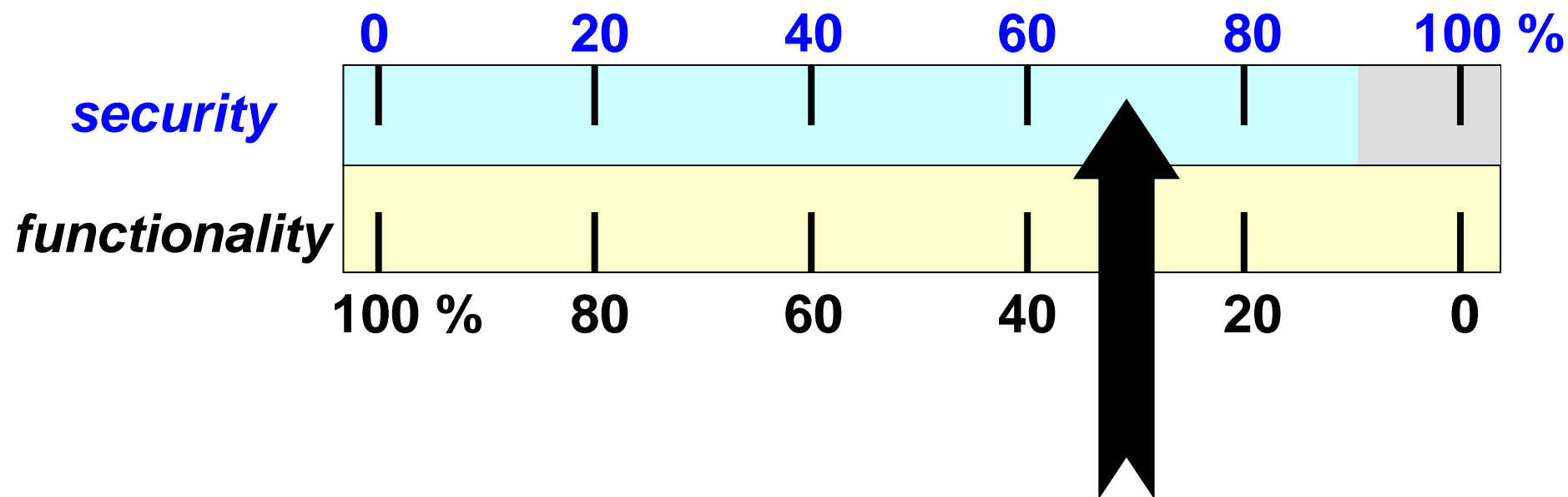
ESFORS Software and Service Development, Security & Dependability Workshop

- complete specification is needed
 - business rule, policy, requirement, ...
- conflicts are possible
 - must be solved (and solution recorded back into specification)
- not a top-down approach
 - rather a backtracking process (trial-and-conflict) ...
 - ... to find a nearly optimal trade-off

Trade-off

ESFORS Software and Service Development, Security & Dependability Workshop

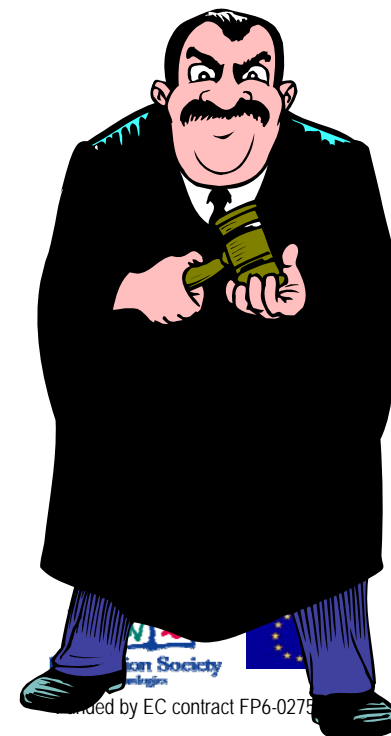
- be ready to give-up some functionality or performance for security



Compliance checking

ESFORS Software and Service Development, Security & Dependability Workshop

- most often asked question today:
“are you compliant with (*insert your favourite para-legal regulation here*)?”
- design, develop and manage an ICT service according to some regulation is not enough
- you must be able to prove it!



Be ready

ESFORS Software and Service Development, Security & Dependability Workshop

- a complete specification helps the auditor
 - e.g. auditing firewall rules (packet-level) with no knowledge of the business rules
- an automatic refinement process helps to demonstrate the absence of errors (but for bugs in the process itself ...)

Dependability

ESFORS Software and Service Development, Security & Dependability Workshop

- as for security, we'll never achieve 100%
(even with a lot of redundancy)
- so what is it about?
- cost-benefit analysis
(or investment done vs. prevented loss)
- ability to predict system behaviour
 - to demonstrate compliance
 - to prepare alternate configurations

System behaviour prediction

ESFORS Software and Service Development, Security & Dependability Workshop

- real system test
 - late!
- emulation
 - reduces complexity, introduces inaccuracy
- simulation
 - needs basic models, pruning of the infinite input space
- formal analysis
 - needs analytical model, must cope with large systems

Computer scientist and engineer

ESFORS Software and Service Development, Security & Dependability Workshop

- be a scientist:
 - new theory must explain known facts, must predict the results of new experiments
 - analyzer must reproduce known behaviour and predict unknown behaviour
- be an engineer:
 - can't design/control a thing that you can't measure
 - define a metrics (different from measure!)
 - do approximations, ignore details

Example

ESFORS Software and Service Development, Security & Dependability Workshop

- MDA
 - model of system, service and business rules
 - refine, design, implement, manage ... always linking to the business rules
- when problem P occurs, alternative configurations C1 and C2 are possible
 - (analysis of C1) 80% of customers working
 - (analysis of C2) 30% of customers working (but this includes 98% of premium users)

Questions?

ESFORS Software and Service Development, Security & Dependability Workshop

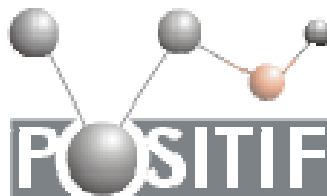
- thanks for your attention



Politecnico di Torino



TORSEC group



POSITIF



DESEREC

