End-to-End Resilience Through External Consistency

Thomas Clédel

2 octobre 2018

What is resilience ?

Resilience Definition

"The persistence of dependability when facing changes." - [Laprie, 2008]

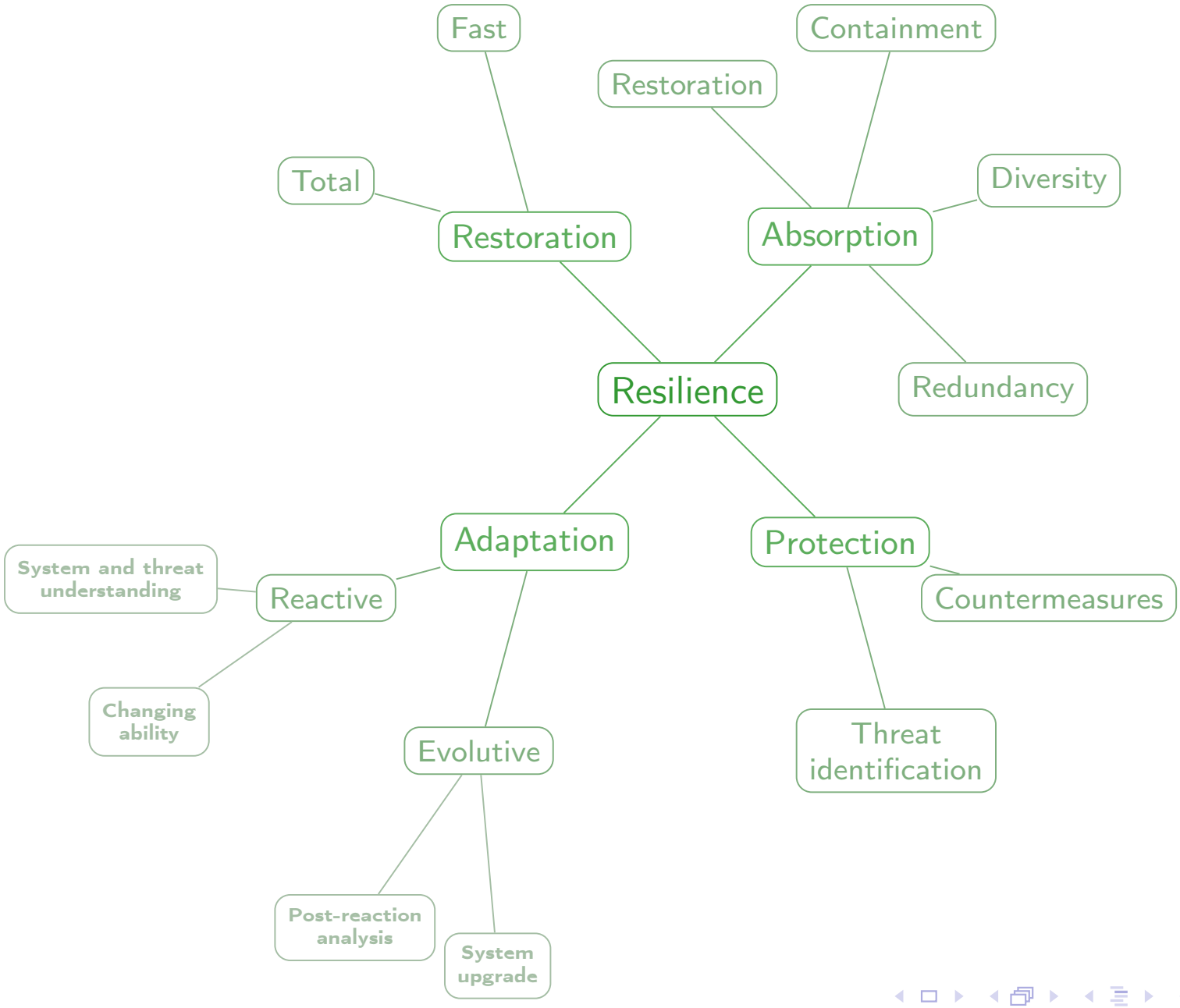
Dependability

- "service delivery that can justifiably be trusted"
- "the avoidance of failures that are unacceptably frequent or severe"

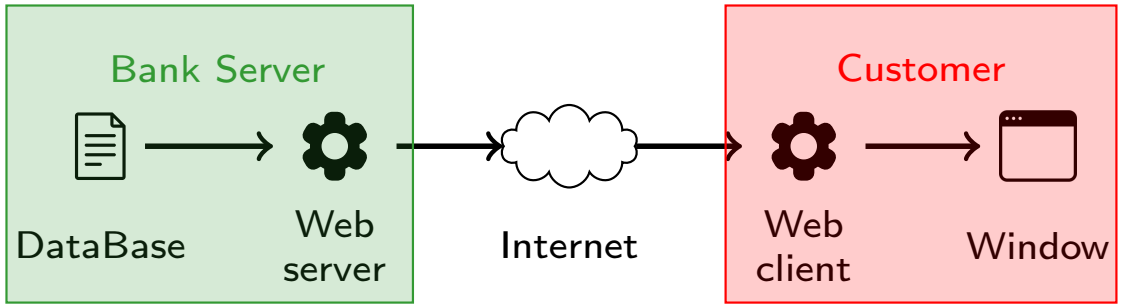
Changes

- intended, unexpected or unknown events
- updates, attacks, faults or failures

A map of resilience



Resilience of a system



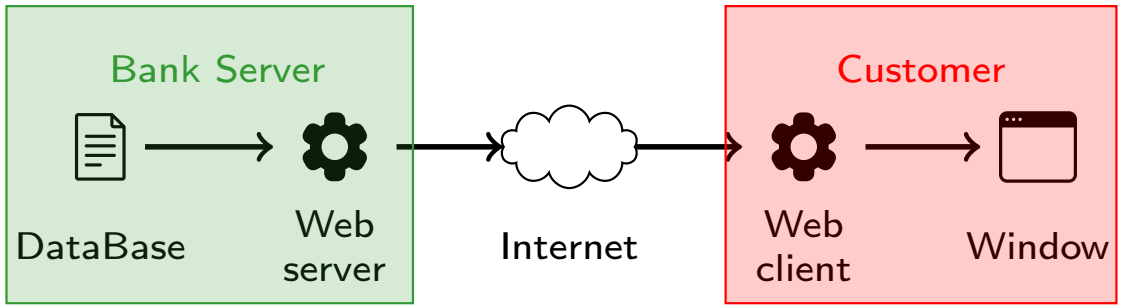
Resilience Definition

"The persistence of service delivery that can justifiably be trusted, when facing changes." - [Laprie, 2008]

System's service

A customer should access its user data

Resilience of a system



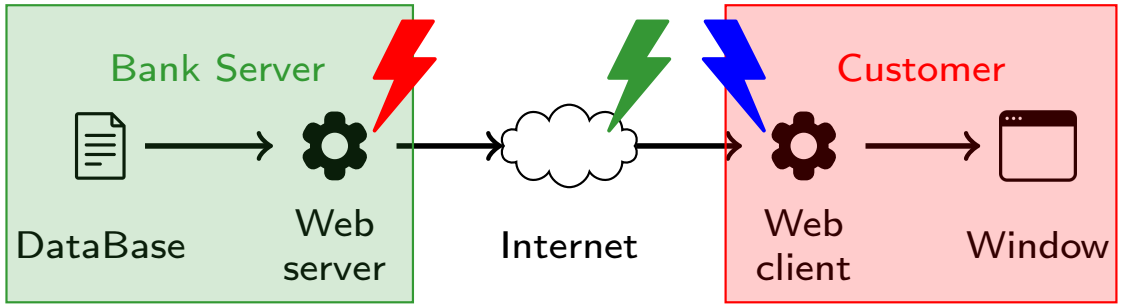
Resilience Definition

"The persistence of service delivery that can justifiably be trusted, when facing changes." - [Laprie, 2008]

System's service

Data shown in the window should be externally consistent with the data stored in the bank database

Possible issues



Issues

Server :

- can be corrupted
- can be compromised

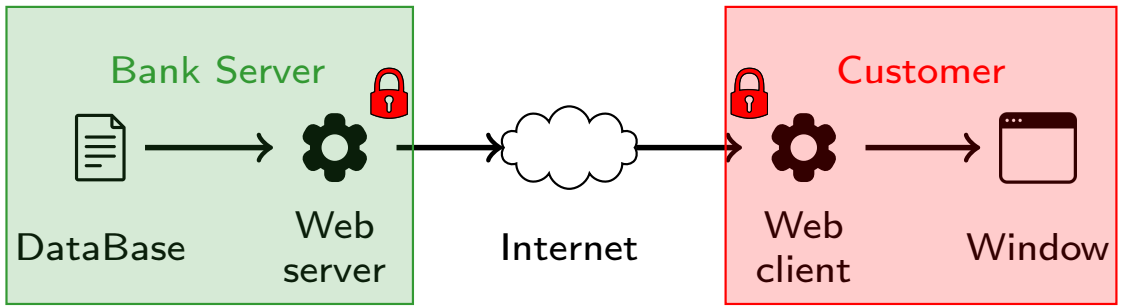
Internet :

- can delete, corrupt, modify received data
- can pretend to be the bank server and send fake data

Client :

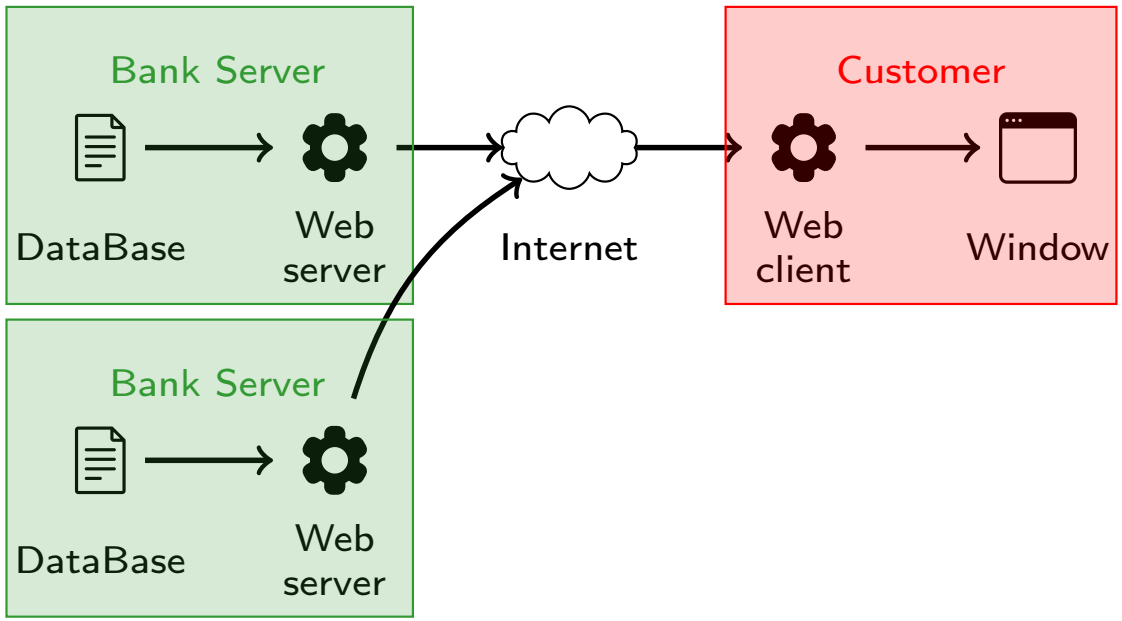
- can be compromised
- can misread received data

Possible remediations



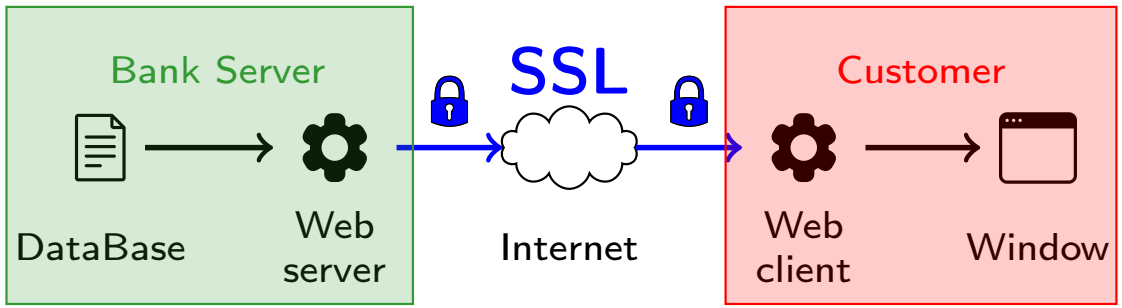
- secured server and/or client
- duplicated server/database
- secured connection

Possible remediations



- secured server and/or client
- duplicated server/database
- secured connection

Possible remediations



- secured server and/or client
- duplicated server/database
- secured connection

A lot of :

- remediations
- system configurations
- security mechanisms
- technologies, etc

A lot of :

- remediations
- system configurations
- security mechanisms
- technologies, etc

Questions

$\Rightarrow$ But which solution is the best to ensure the system's service delivery ?

A lot of :

- remediations
- system configurations
- security mechanisms
- technologies, etc

Questions

$\Rightarrow$ But which solution is the best to ensure the system's service delivery ?

$\Rightarrow$ How much is a solution better than the others ?

A lot of :

- remediations
- system configurations
- security mechanisms
- technologies, etc

Questions

$\Rightarrow$ But which solution is the best to ensure the system's service delivery ?

$\Rightarrow$ How much is a solution better than the others ?

$\Rightarrow$ How much does a solution improve the resilience of a system ?

End-to-End Resilience through External Consistency

Main principles

Principle 1

System services = system output data

External Consistency Definitions

- "the correspondence between the data object and the real world object it represents" - [Clark and Wilson, 1987]
- "the ability of a computing system to give correct information about its external environment" - [Williams and La Padula, 1993]

Principle 2

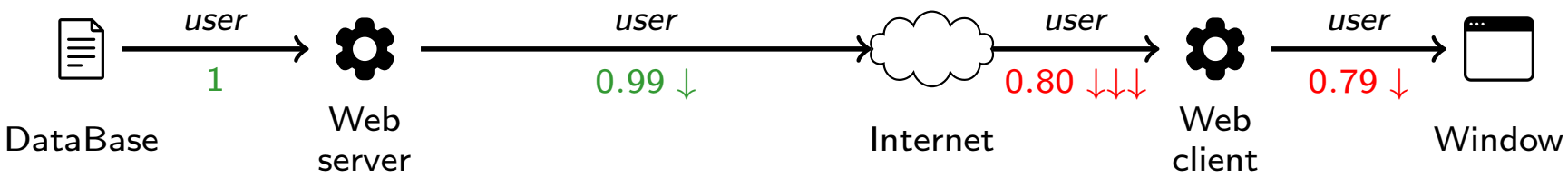
Service delivery = the corresponding data are likely to be externally consistent

End-to-End Resilience through External Consistency

Our Model

Model Elements / Basic Ideas

- system = oriented graph
  $\Rightarrow$ vertices = components, edges = relationships
- system services $\approx$ data dimensions / data attributes
- components $\hookleftarrow$ resilience functions
- system resilience $\approx$ likelihood of external consistency of data attributes

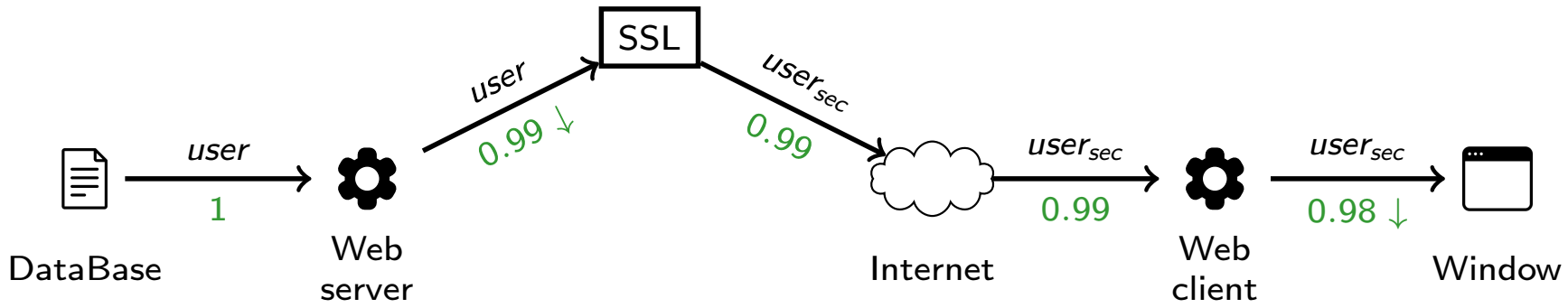


End-to-End Resilience through External Consistency

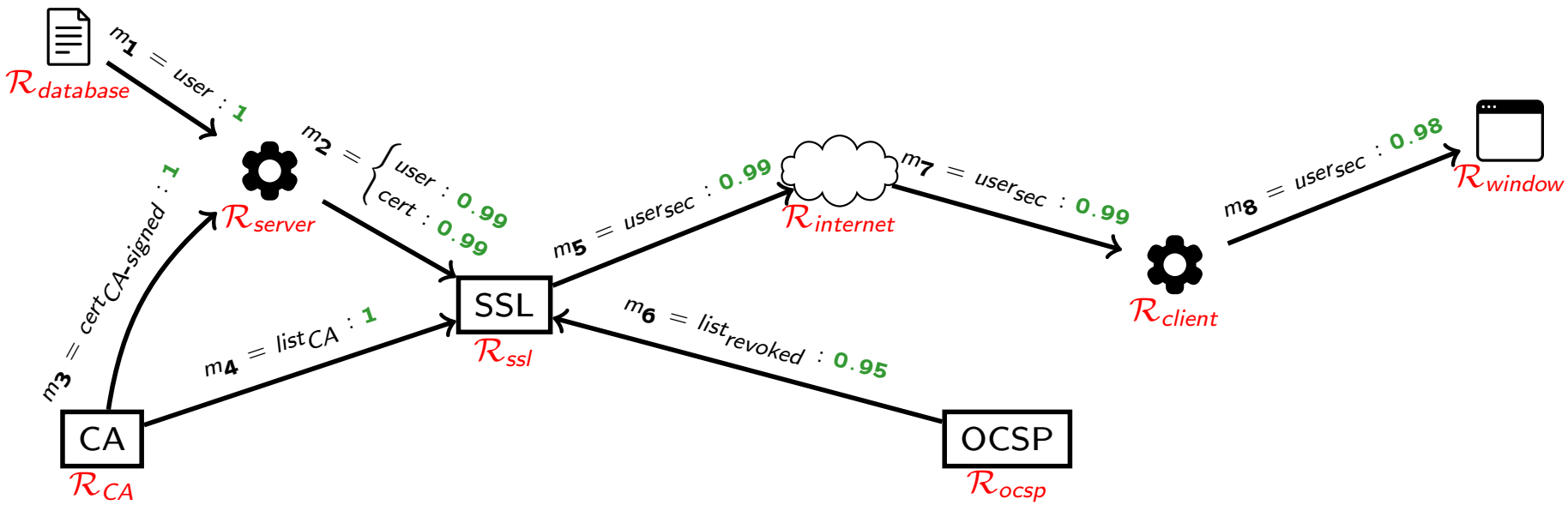
Our Model

Model Elements / Basic Ideas

- system = oriented graph
  $\Rightarrow$ vertices = components, edges = relationships
- system services $\approx$ data dimensions / data attributes
- components $\hookleftarrow$ resilience functions
- system resilience $\approx$ likelihood of external consistency of data attributes



A more complex use-case

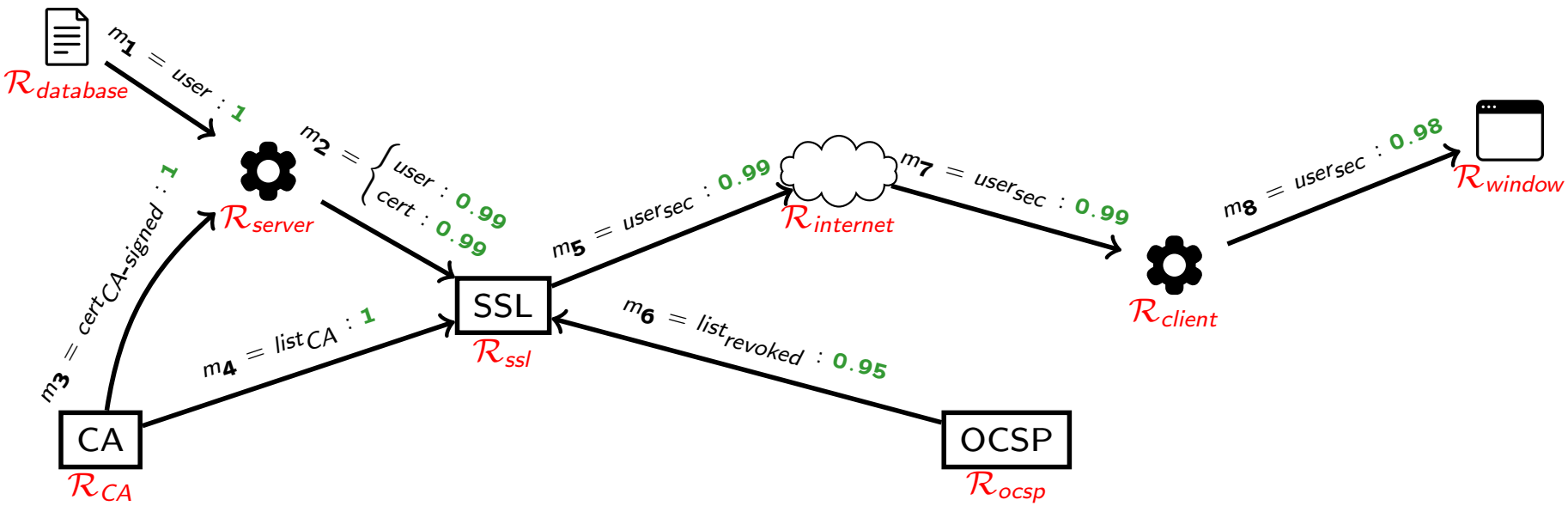


m_1, m_3, m_4, m_6 : initial external consistency values for data attributes

$$m_2 = \mathcal{R}_{server}(m_1, m_3) \qquad m_5 = \mathcal{R}_{ssl}(m_2, m_4, m_6)$$
$$m_7 = \mathcal{R}_{internet}(m_6) \qquad m_8 = \mathcal{R}_{client}(m_7)$$

System definition - $sys = \langle V, E, L, \mathcal{R} \rangle$

Directed acyclic graph



Vertices

$V = S + C = Sources + OthersComponents$

$\rightarrow S = \{database, CA, OCSP\}$

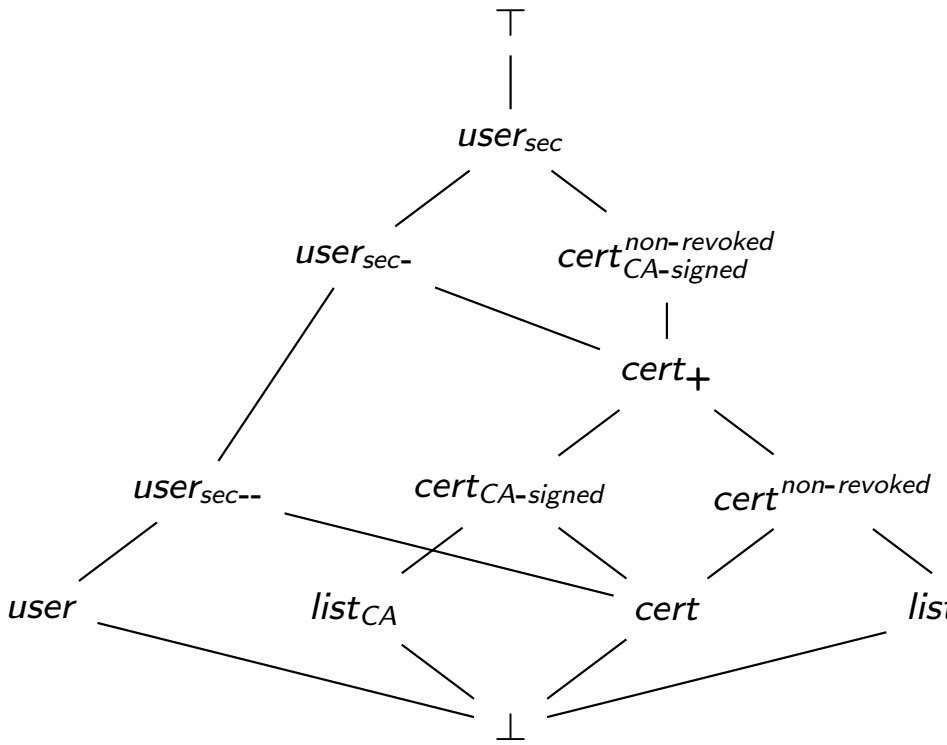
$\rightarrow C = \{server, ssl, internet, client, window\}$

Edges

$E = \{(database, server), (CA, server), \ldots, (client, window)\}$

System definition - $sys = \langle V, E, L, \mathcal{R} \rangle$

Ordered data dimensions



$D = \{user, list_{CA}, cert, list_{revoked}, \ldots, user_{sec}\}$

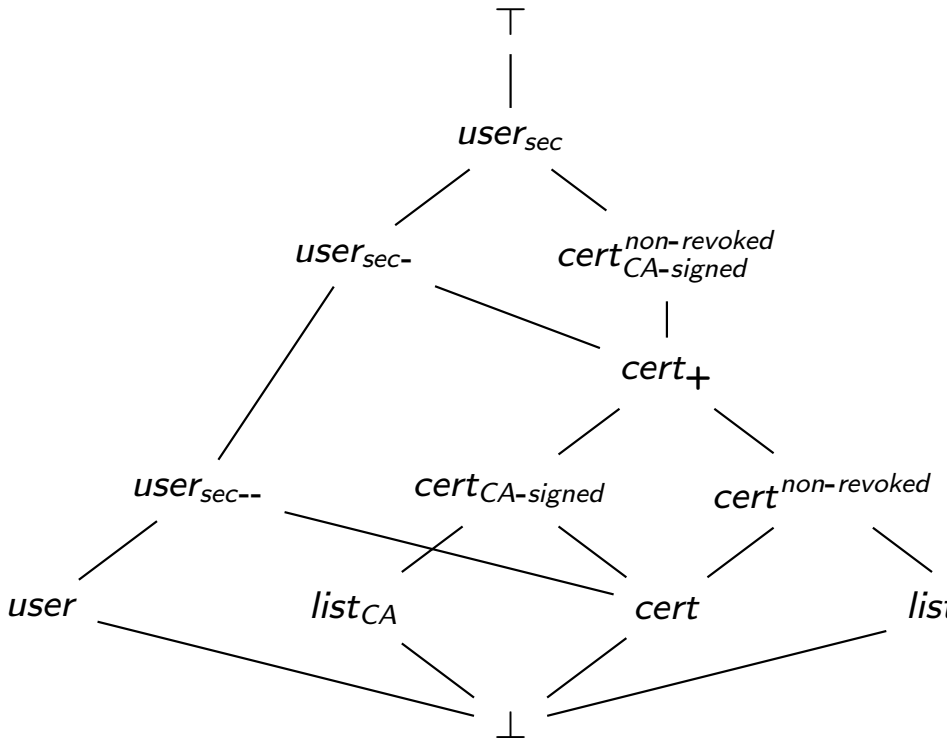
user: user data
$list_{CA}$: list of CA certificates
cert: certificate
$list_{revoked}$: list of revoked certificates
$user_{sec}$: same as *user* but using *ssl*

Lattice of data dimensions

$L = (D, \leq)$

System definition - $sys = \langle V, E, L, \mathcal{R} \rangle$

Ordered data dimensions



$D = \{user, list_{CA}, cert, list_{revoked}, \ldots, user_{sec}\}$

user: user data
$list_{CA}$: list of CA certificates
cert: certificate
$list_{revoked}$: list of revoked certificates
$user_{sec}$: same as *user* but using *ssl*

Constraint on Data vectors/mappings

$$\mathcal{D} = \left\{ m : D \to [0,1] \mid \forall d, m(d) \leq \max_{\substack{d' \in D, d' \leq d \\ d' \neq d, d' \neq \bot}} (m(d')) \right\}$$

System definition - $sys = \langle V, E, L, \mathcal{R} \rangle$

Resilience functions of components

Component transformation

$$\Theta = \left\{ \begin{array}{l} \tau : (V \nrightarrow DATA) \nrightarrow (V \nrightarrow DATA) \mid \\ \quad \exists! v_d \in V \mid dom(\tau) = \{M \mid cond_1 \wedge cond_2 \wedge cond_3\} \end{array} \right\}$$

with $cond_1 : dom(M) = \{v_s \mid (v_s, v_d) \in E\}$

$cond_2 : dom(\tau(M)) = \{v_d\}$

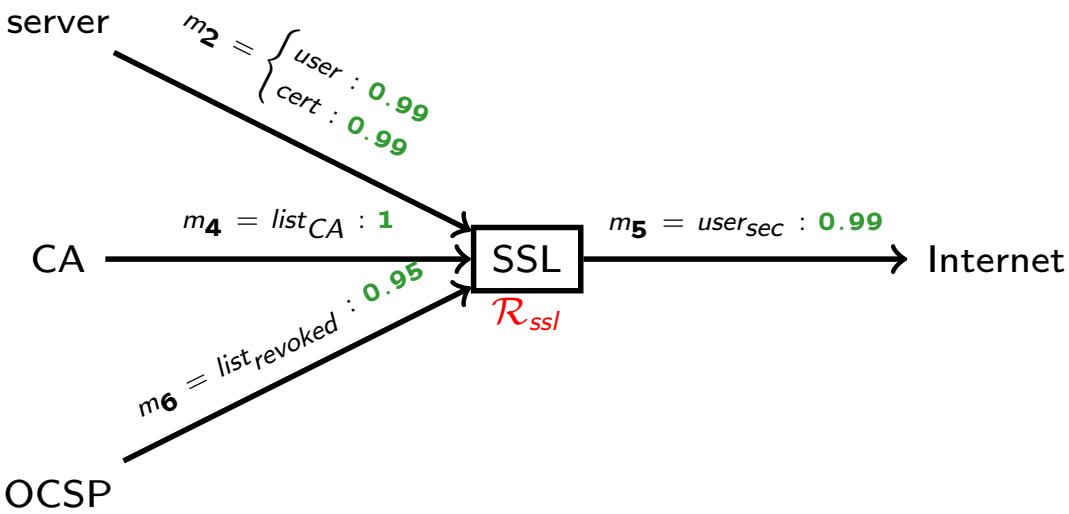
$$cond_3 : \forall d \in D, \tau(M)(v_d)(d) \leq \max_{\substack{d' \in D \\ d' \leq d, d' \neq \bot}} \left(\bigoplus_{v_s \in dom(M)} M(v_s)(d') \right)$$

Resilience of components

$\mathcal{R} : V \to (C \nrightarrow \Theta)$

System definition - $sys = \langle V, E, L, \mathcal{R} \rangle$

Resilience functions of components



$$\mathcal{R}_{ssl}\left(Internet\right)\left(\left\{\left(server, m_2\right), \left(CA, m_4\right), \left(OCSP, m_6\right)\right\}\right) = \left\{\left(Internet, m_5\right)\right\}$$

Conclusion

Evaluation of resilience

- through external consistency likelihood of data
- system services $\approx$ lattice of data dimensions
- components manipulate input data through dimensions and likelihoods
  $\rightarrow$ constrained by the data dimensions lattice
- limitations : no cycles allowed

Further steps

- more detailed component behaviour
- remove limitations
- automatic conversion from system services into data dimensions
- integration of resilience capacities (adaptation, restoration) into the model
- ...

David D. Clark and David R. Wilson. A comparison of commercial and military computer security policies. In *Security and Privacy, 1987 IEEE Symposium on*, pages 184–184. IEEE, 1987. URL `http://ieeexplore.ieee.org/abstract/document/6234890/`.

Jean-Claude Laprie. From dependability to resilience. In *38th IEEE/IFIP Int. Conf. On Dependable Systems and Networks*, pages G8–G9. Citeseer, 2008.

James G. Williams and Leonard J. La Padula. Automated support for external consistency. In *Computer Security Foundations Workshop VI, 1993. Proceedings*, pages 71–81. IEEE, 1993. URL `http://ieeexplore.ieee.org/abstract/document/246637/`.