SERENE 2014: Panel

Views on Runtime Resilience Assessment of Dynamic Software Systems

SERENE
Oct. 2014

Marco Vieira

mvieira@dei.uc.pt

University of Coimbra – Portugal

Context

- **Motivation**: Software and software-based systems are becoming extremely complex
  - Extremely difficult to ensure resilience!!!
  - Increasing use of software in evolvable and (safety-, mission-, business-) critical systems
- **Definition**: Resilience…
  - Ability of a system to persistently deliver its services in a dependable way…
  - … when facing changes, failures and intrusions
- **Need**: New methods and tools for assessing the resilience of software systems at runtime

Objective

Present and discuss views
on the future needs and techniques for
runtime resilience assessment
of *dynamic* software systems

- Joining a set of well-known experienced panelists in different domains:
  - Resilience assessment
  - Software engineering
  - Dynamic systems design
  - Dependable computing
  - …

Who?

- Elena Troubitsyna, Åbo Akademi Univ., Finland
- Katinka Wolter, Freie University Berlin, Germany
- Vincenzo De Florio, Univ. of Antwerp, Belgium
- Henry Muccini, University of L'Aquila, Italy
- Alexander Romanovsky, Newcastle Univ., UK

- Marco Vieira, Univ. of Coimbra, Portugal

Key aspects…

- Metrics to characterize resilience
- Definition of dynamic workloads and of *changeloads*
- Runtime monitoring of dynamic and unbounded systems
- Runtime modeling and experimentation
- Dissemination, training, and standardization

Outline

- Brief views
- Discussion

SERENE 2014: Panel

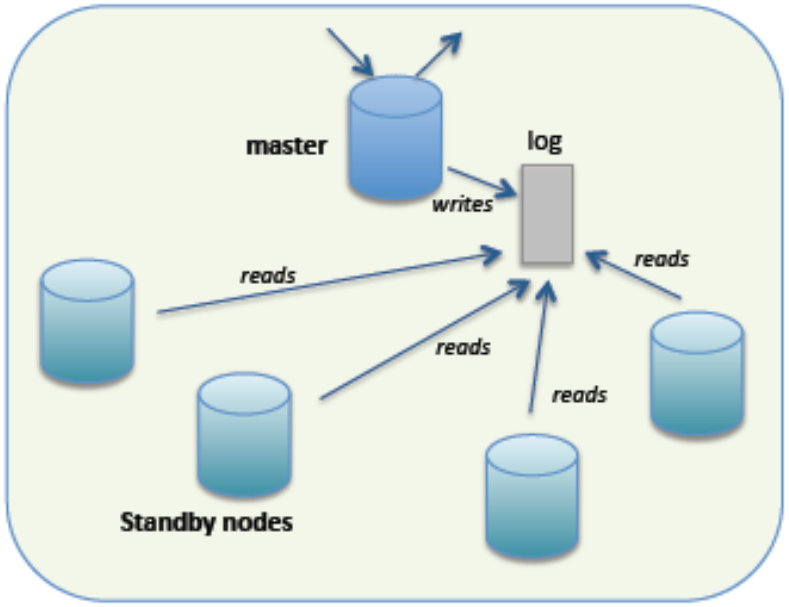
Views on Runtime Resilience Assessment of Dynamic Software Systems

SERENE
Oct. 2014

Elena Troubitsyna
Åbo Akademi University, Finland

Dynamic Software Systems: challenges

- Worldwide consumer digital storage needs will grow from 329 exabytes in 2011 to 4.1 zettabytes in 2016 (Gartner)
- Cloud data store
  - Massive replication
  - Write ahead logging
- How to avoid resource over-provisioning or underprovisioning?



Need for proactive resilience

- Monitoring: what and how much?
- Prediction: how to learn trends, choose or synthesise adaptation strategy?
- Autonomous adaptation: how to verify?
- State-of-the-art: autonomic computing
- State-of-the-practice: manual human monitoring and adaptation
- Challenge: bringing research to practice
  - Demonstrators, guidelines, cookbooks

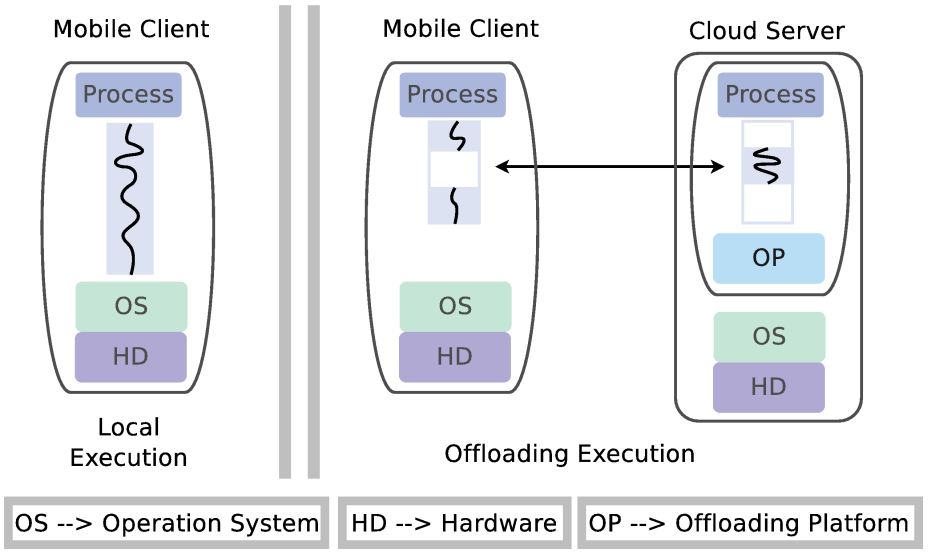
SERENE 2014: Panel

Views on Runtime Resilience Assessment of Dynamic Software Systems

SERENE
Oct. 2014

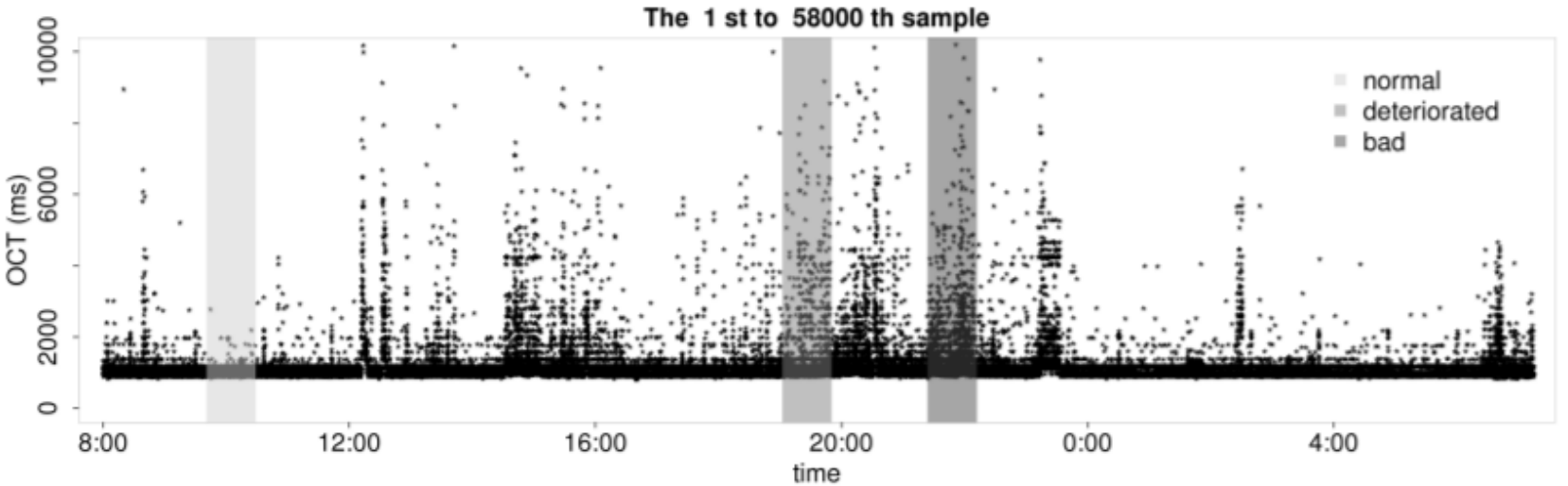
Katinka Wolter
Freie University Berlin, Germany

Views on Runtime Resilience Assessment of Dynamic Software Systems



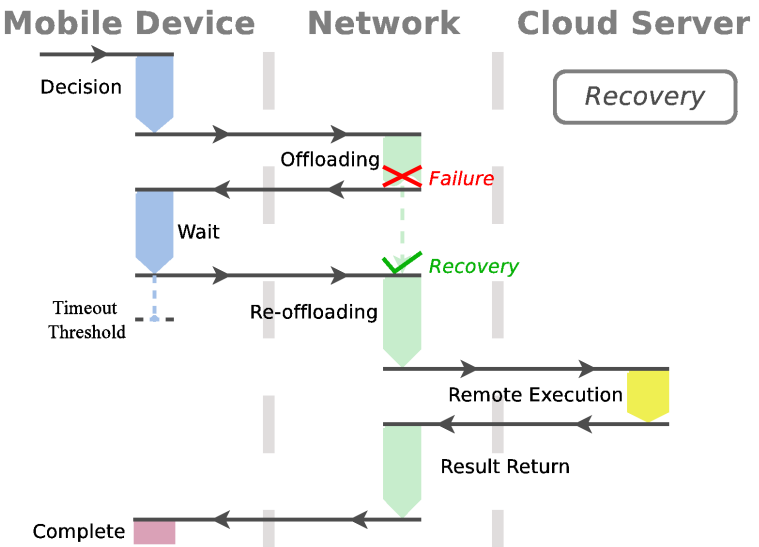
- Mobile offloading requires resilient environment

Views on Runtime Resilience Assessment of Dynamic Software Systems



- Mobile offloading requires resilient environment
- Observed conditions constantly vary

Views on Runtime Resilience Assessment of Dynamic Software Systems



- Mobile offloading requires resilient environment
- Observed conditions constantly vary
- Restart decisions and timeout must be adaptively based on observations - how long to wait?

Adaptivity

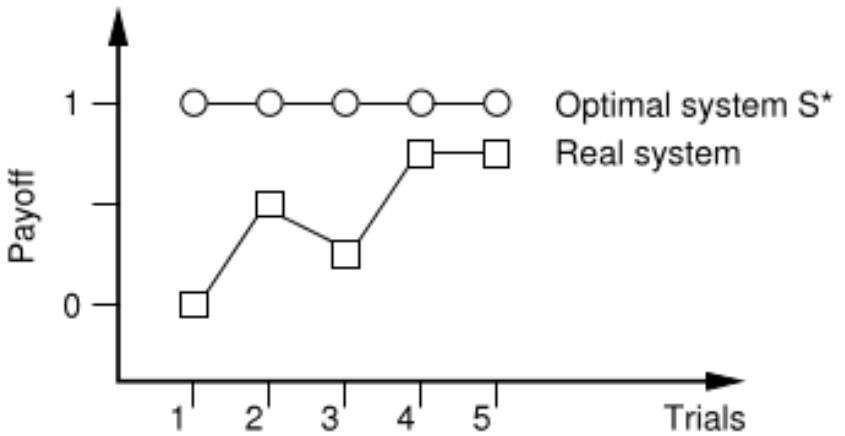


Fig. 3. Example of adaptive system behaviour.

- For adaptivity need to know the optimal system behaviour and payoff

Adaptivity

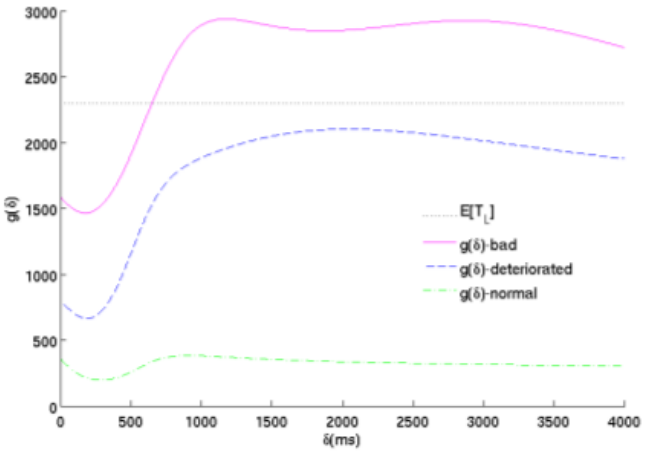


Figure 8: Restart timeout for the different subsets of the data

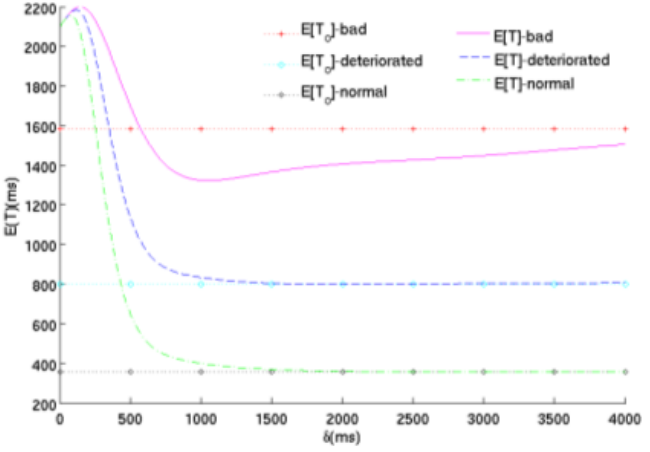


Figure 9: Expectation of OCT with/without the local restart versus τ

- For adaptivity need to know the optimal system behaviour and payoff
- Change the timeout such that metric is improved

Adaptivity

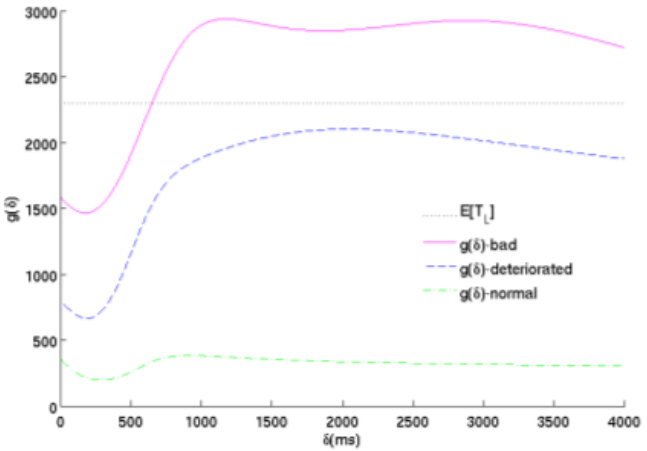


Figure 8: Restart timeout for the different subsets of the data

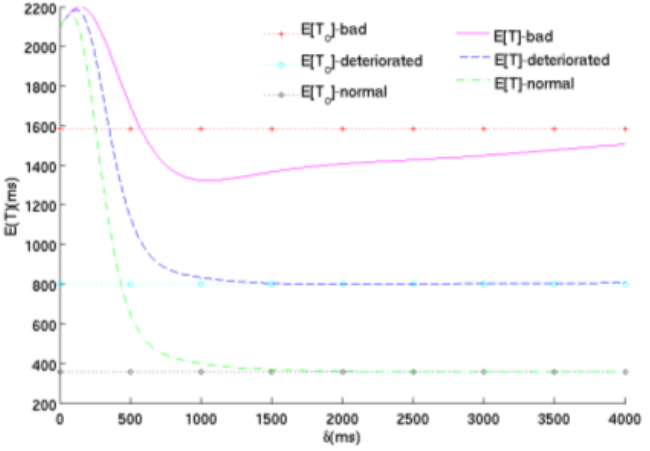


Figure 9: Expectation of OCT with/without the local restart versus τ

- For adaptivity need to know the optimal system behaviour and payoff
- Change the timeout such that metric is improved
- Ability to change is adaptivity

SERENE 2014: Panel

Views on Runtime Resilience Assessment of Dynamic Software Systems

SERENE
Oct. 2014

Vincenzo De Florio
University of Antwerp, Belgium

Resilience ← Aristotelian entelechy

Resilience ← Aristotelian entelechy

"Being-at-work staying-the-same"

Resilience ← Aristotelian entelechy

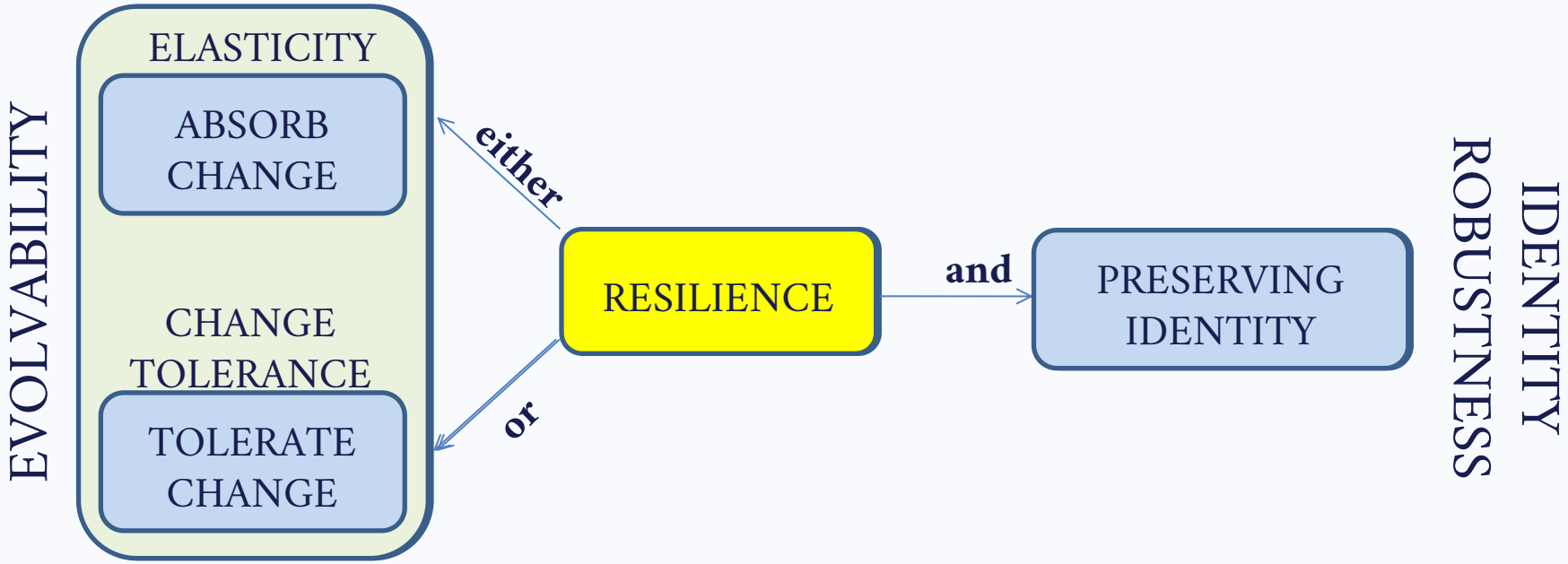
"Being-at-work staying-the-same"

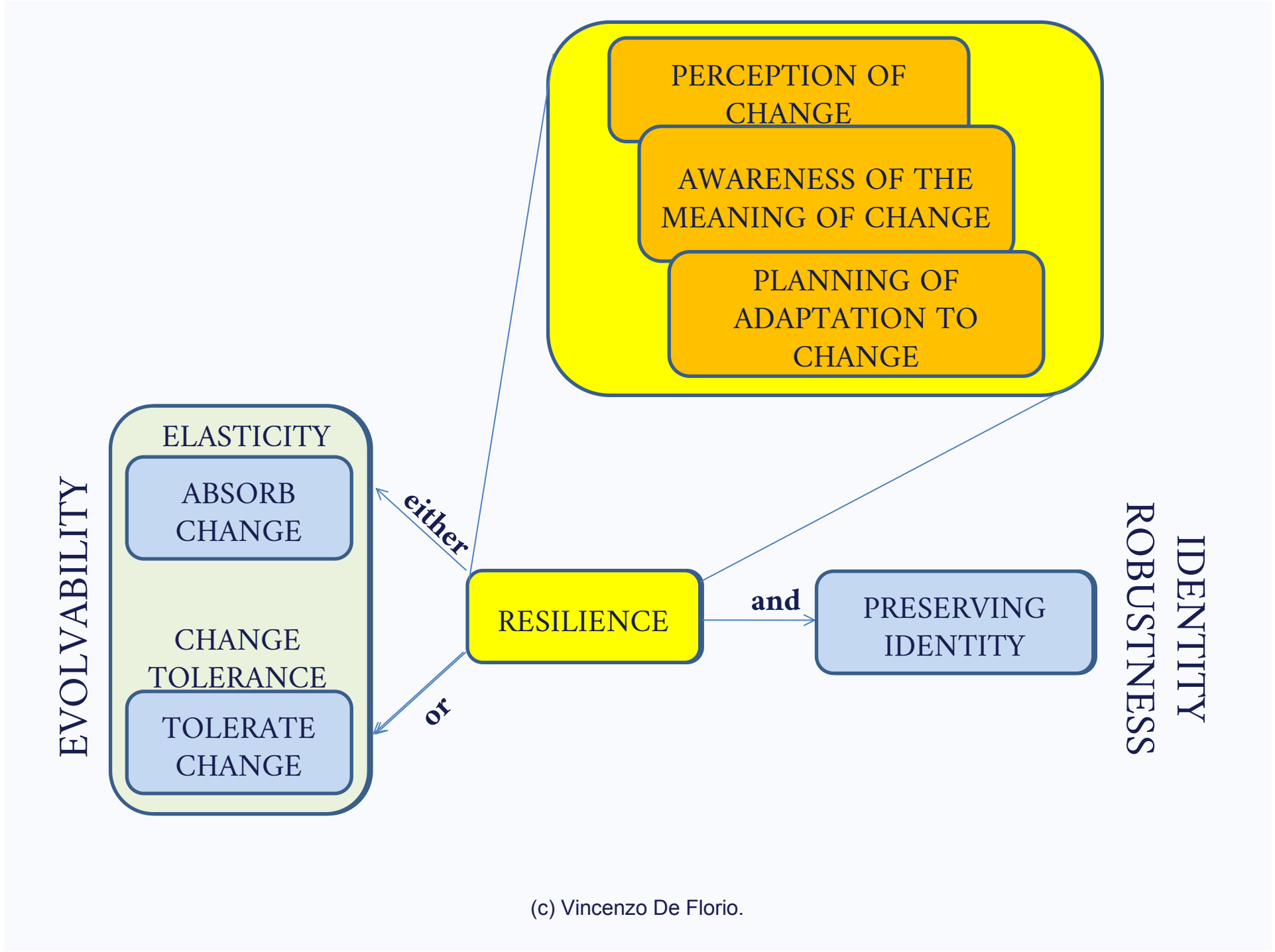
EVOLVABILITY

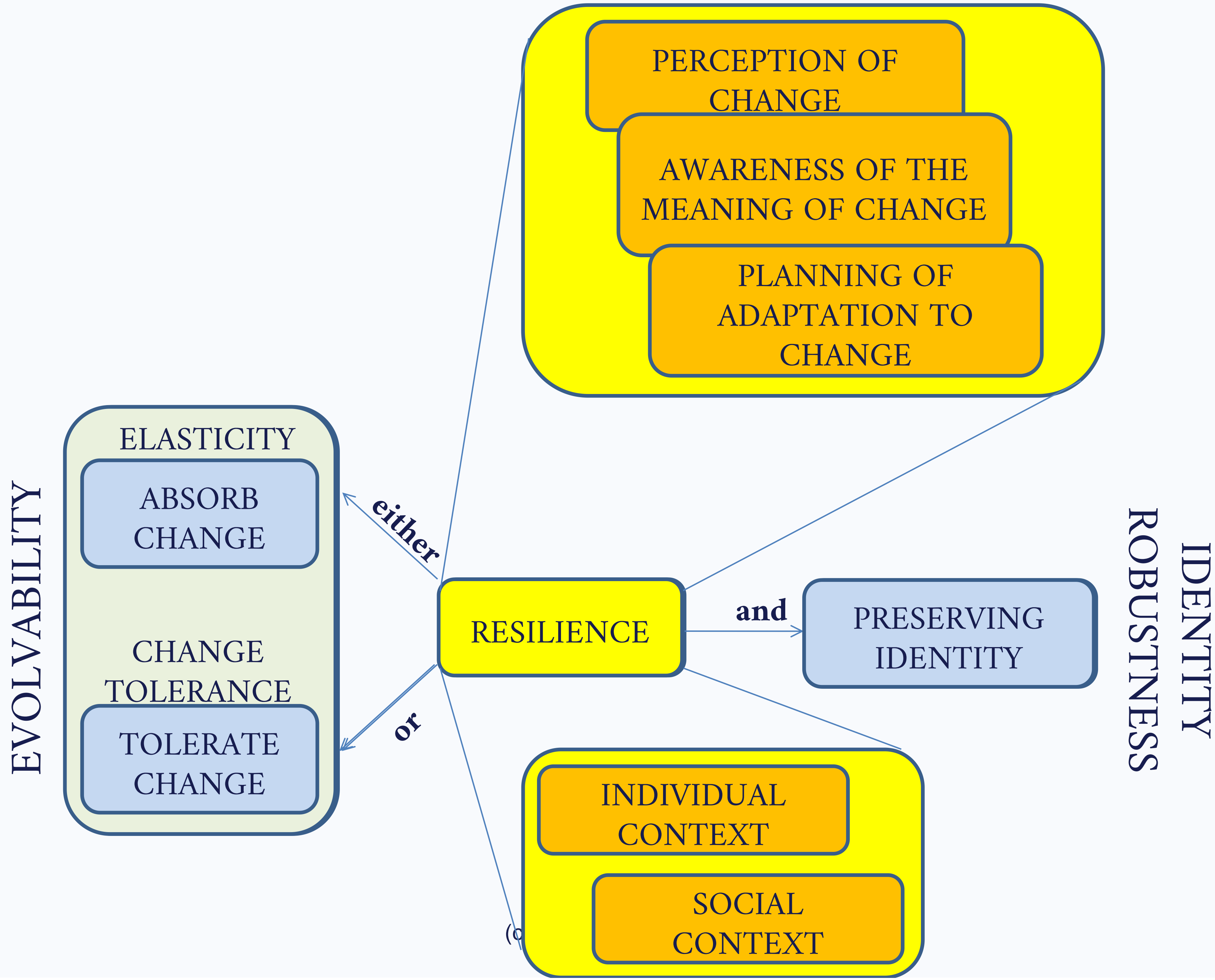
IDENTITY
ROBUSTNESS

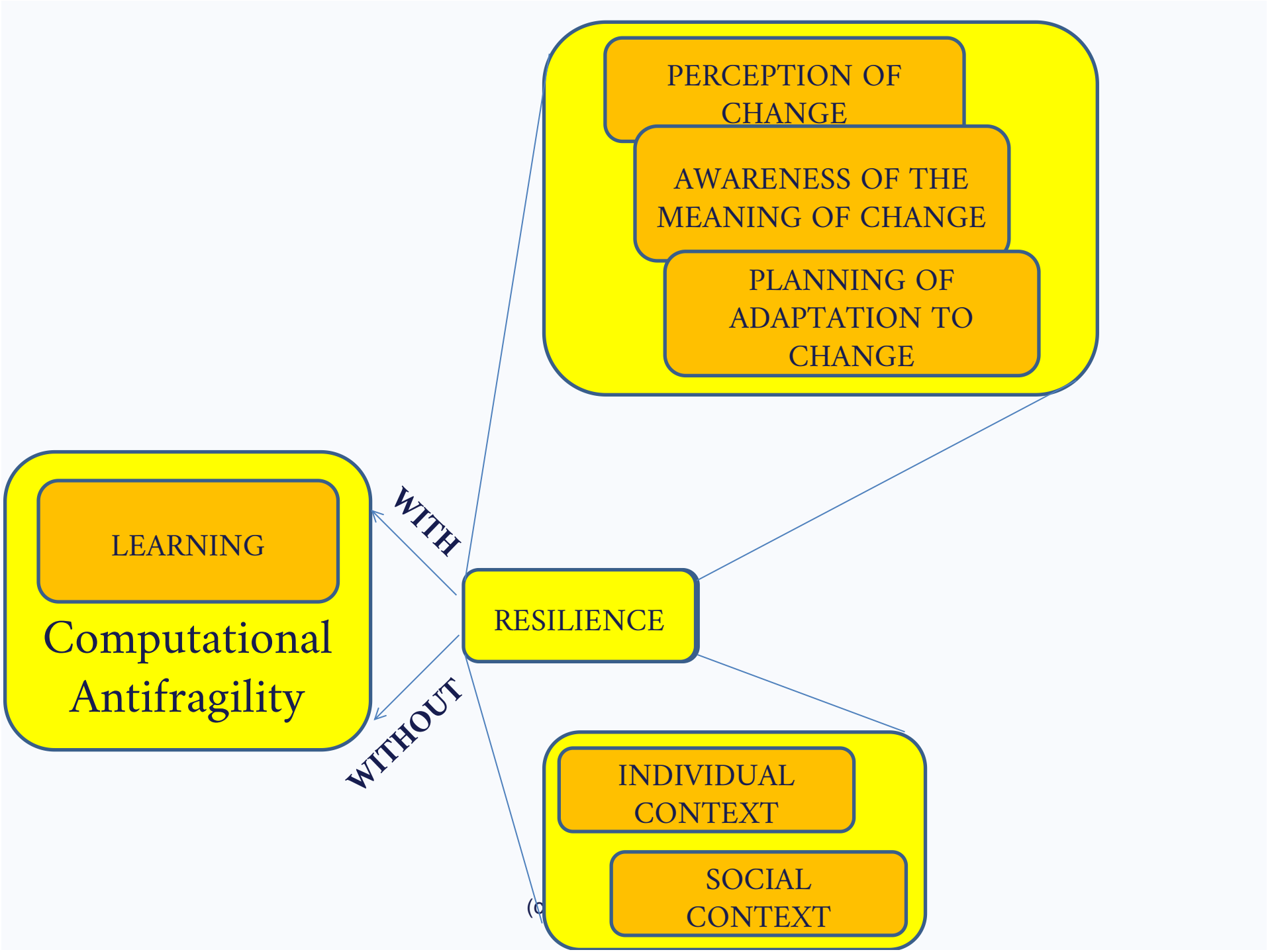
Resilience ← Aristotelian entelechy

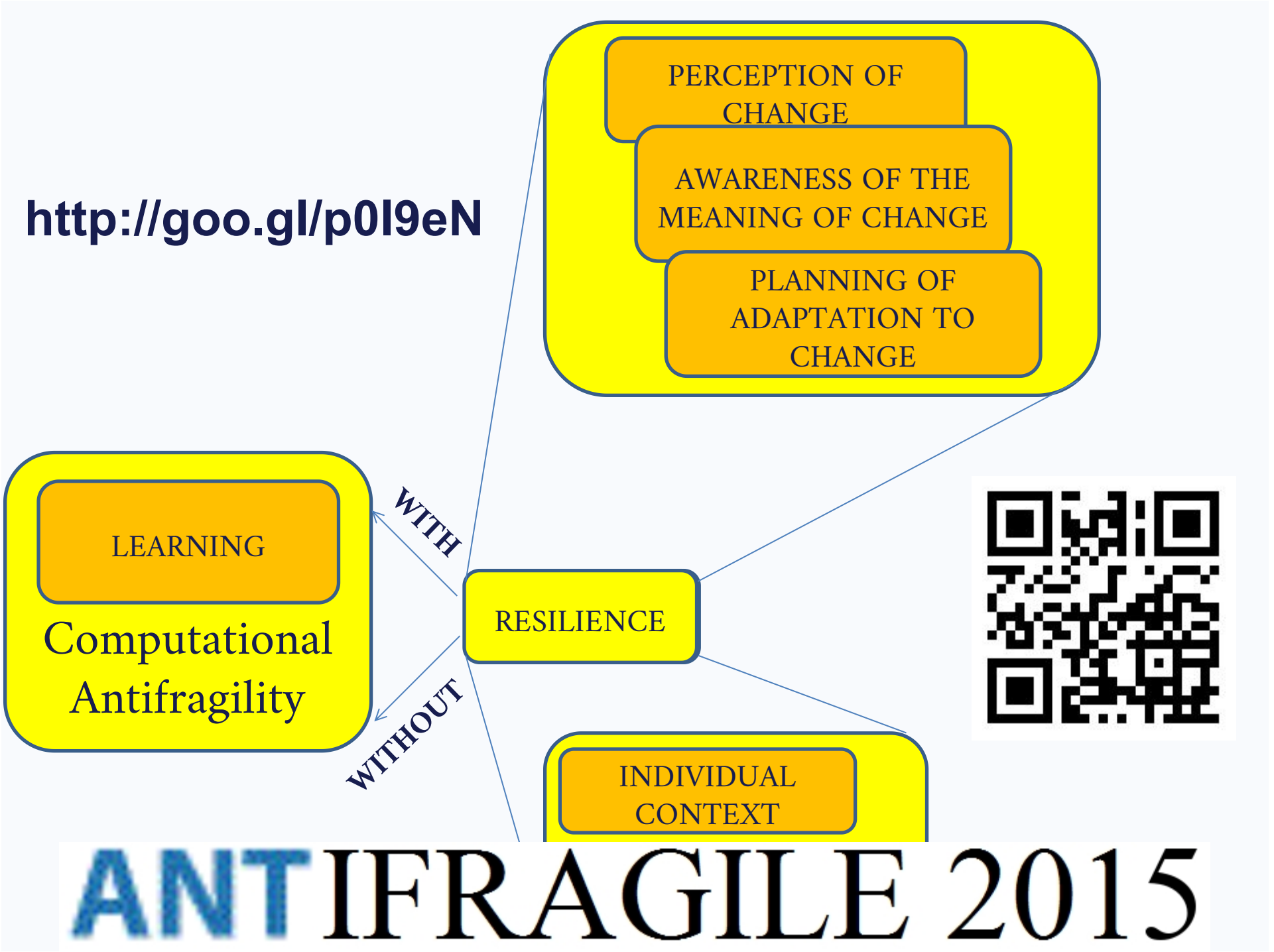
"Being-at-work staying-the-same"






PERCEPTION OF CHANGE
AWARENESS OF THE MEANING OF CHANGE
PLANNING OF ADAPTATION TO CHANGE
EVOLVABILITY
ELASTICITY
ABSORB CHANGE
CHANGE TOLERANCE
TOLERATE CHANGE
either
or
RESILIENCE
and
PRESERVING IDENTITY
IDENTITY ROBUSTNESS
INDIVIDUAL CONTEXT
SOCIAL CONTEXT


PERCEPTION OF CHANGE
AWARENESS OF THE MEANING OF CHANGE
PLANNING OF ADAPTATION TO CHANGE
LEARNING
Computational Antifragility
WITH
WITHOUT
RESILIENCE
INDIVIDUAL CONTEXT
SOCIAL CONTEXT


http://goo.gl/p0l9eN
PERCEPTION OF CHANGE
AWARENESS OF THE MEANING OF CHANGE
PLANNING OF ADAPTATION TO CHANGE
LEARNING
Computational Antifragility
WITH
WITHOUT
RESILIENCE
INDIVIDUAL CONTEXT
ANTIFRAGILE 2015

SERENE 2014: Panel

Views on Runtime Resilience Assessment of Dynamic Software Systems

SERENE
Oct. 2014

Henry Muccini
University of L'Aquila, Italy

Runtime Monitoring of Dynamic Systems: strategy

- Combine run-time and design-time info
- "Proactive" monitoring for fault prediction
- Monitored properties may evolve themselves

F. De Angelis, A. Polini, H. Muccini and M. R. Di Berardini, «CASSANDRA - an Online Failure Prediction Strategy for Dynamically Evolving Systems», in ICFEM 2014

"Views on Runtime Resilience Assessment of Dynamic Software Systems" panel @SERENE2014

Henry Muccini
University of L'Aquila, Italy
disim

Runtime Monitoring of Dynamic Systems: challenges

- Everything may (need to) evolve!
- Technical infrastructure
- Evolve fast, monitor fast

"Views on Runtime Resilience Assessment of Dynamic Software Systems" panel @SERENE2014

Henry Muccini
University of L'Aquila, Italy
disim

SERENE 2014: Panel

Views on Runtime Resilience Assessment of Dynamic Software Systems

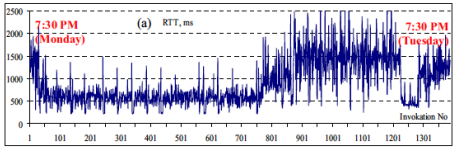
SERENE
Oct. 2014

Alexander Romanovsky
Newcastle University, UK

Internet/cloud-based/SOA systems

- Dealing with uncertainty and ensuring resilience
- Dynamic/adaptive fault tolerance
  - choice of timeouts, replica locations/numbers, workflow patterns
- Monitoring (off-line and dynamic assessment)
  - Probabilistic nature of events (distributions)
  - Prediction
- Non-ACID DBs: the CAP theorem about trading off consistency, availability and message loss/ partition failures
  - Timeout is trading off availability vs. consistency

Newcastle University



Many-core systems

- The PEAR triangle – performance vs. energy consumption vs. reliability
- Layers: application, OS, HW
- Adaptive management: sensors and actuators
- What, where and how we measure (detection)
  - temperature sensors on chips, cycles, energy consumption, load, delays/time-outs/missed deadlines, deterioration of the QoS
- What do we do (recovery)
  - modes, dynamic frequency/voltage scaling, reconfiguration, number of replicas/reconfiguration, less precise computations

Your Time!

Don’t run from my questions…

- What makes a good resilience metric?
  - Examples of metrics
  - Representation
- Definition of dynamic workloads and of *changeloads*
  - What is a changeload?
  - What changes?
  - Is the workload part of the changeload?

Don’t run from my questions…

- Runtime monitoring of dynamic and unbounded systems
  - Unbounded? What is unbounded?
  - Resilience of the monitors?
- Runtime modeling and experimentation
  - How to maintain accurate models at runtime?
  - How to run experiments in runtime environments?
  - How is this related to runtime monitoring?
- Dissemination, training, and standardization
  - What can/should we do here?

Elena

- Cloud data store: how to avoid under-provisioning and over-provisioning?
  - Continuous adaptation
  - SLAs
- What to monitor?
  - Source to feed runtime adaptation
- Needs:
  - Prediction, not just detection
  - Continuous verification
- Key Question: How to bring research to practice?

Katinka

- Mobile offloading
  - Requires a resilient environment
- How to know the optimal system behavior?
  - Monitor!
- How well systems adapt?
  - Monitor!
- Needs:
  - Prediction, not just detection
  - Adaptability – how to measure?

Vincenzo

- Key concepts: elasticity + change tolerance
  - Elasticity – Design
  - Change tolerance – runtime
  - Resilience = elasticity and/or change tolerance?
- Systems can be considered in isolation or under a social context
  - Context is very relevant…
- Resilience with a backup learning process
  - Anti-fragility…
- Anti-fragility: anticipation / prediction

Henry

- Runtime monitoring of dynamic systems
- Combine runtime + design information
- Monitored properties may evolve
  - What about requirements?
- Consider the dynamicity of the monitors
  - Go after the system evolution
- How to characterize *sensivity* properties

Sascha

- Internet / Cloud / SOA
- Challenge: Dealing with uncertainty
  - Prediction? Short time only?
- We need data
  - How to handle this data?
- Is resilience about a trade-off?
- Future: Performance vs Energy vs Reliability
- Good enough SW engineering…

...

- All talked about prediction
  - Is it really essential for Resilient systems?
- Systems must be "opened" for monitoring?
- What about resileince under security attacks?
- Known unknown vs Unknown unknown