Providing Resilience and Efficiency to the Internet of Things

Ingrid Nunes
Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre, Brazil

SEfIAS GI-Dagstuhl Seminar, 2018

.inf INSTITUTO DE INFORMÁTICA UFRGS

Prosoft

UFRGS UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

About me

- Academic at UFRGS (Federal University of Rio Grande do Sul, BR) since 2012
  - Design/Implementation/Evolution Problems
    - ZAPALOWSKI, V. ; NUNES, I. ; NUNES, D. The WGB **Method to Recover Implemented Architectural Rules**. Information and Software Technology, 2018.
  - User Agents/Decision Making (PhD)
    - NUNES, I. ; MILES, S. ; LUCK, M. ; BARBOSA, S. D. J. ; Lucena, Carlos J. P. **Decision Making with Natural Language based Preferences and Psychology-inspired Heuristics**. Engineering Applications of Artificial Intelligence, v. 42, p. 16-35, 2015.
    - NUNES, I., MILES, S., LUCK, M., BARBOSA, S.D.J., LUCENA, C. **Pattern-based Explanation for Automated Decisions**, ECAI 2014.
    - Explanations in Recommender Systems (2016/2017 - sabbatical Year at TU Dortmund, DE)
      - NUNES, I.; JANNACH, D. **A systematic review and taxonomy of explanations in decision support and recommender systems**. User Model and User-Adapted Interaction, 2017.

About me

- Agent-oriented Software Engineering
  - Customisations of BDI Agents
    - NUNES, I.; LUCK, M. **Softgoal-based Plan Selection in Model-driven BDI Agents**, AAMAS 2014
    - FACCIN, J. G. ; NUNES, I. **A tool-supported development method for improved BDI plan selection**. Engineering Applications of Artificial Intelligence, 2017.
  - Agent-based Simulation
    - SANTOS, F. ; NUNES, I. ; BAZZAN, A. L. C. **Model-driven agent-based simulation development: A modeling language and empirical evaluation in the adaptive traffic signal control domain**. Simulation Modelling Practice and Theory, 2018.
  - Computer Networks
    - SCHARDONG, F. ; NUNES, I. ; SCHAEFFER FILHO, A.E. **Providing Cognitive Components with a Bidding Heuristic for Emergent NFV Orchestration**. NOMS, 2018.
- … and self-adaptive systems

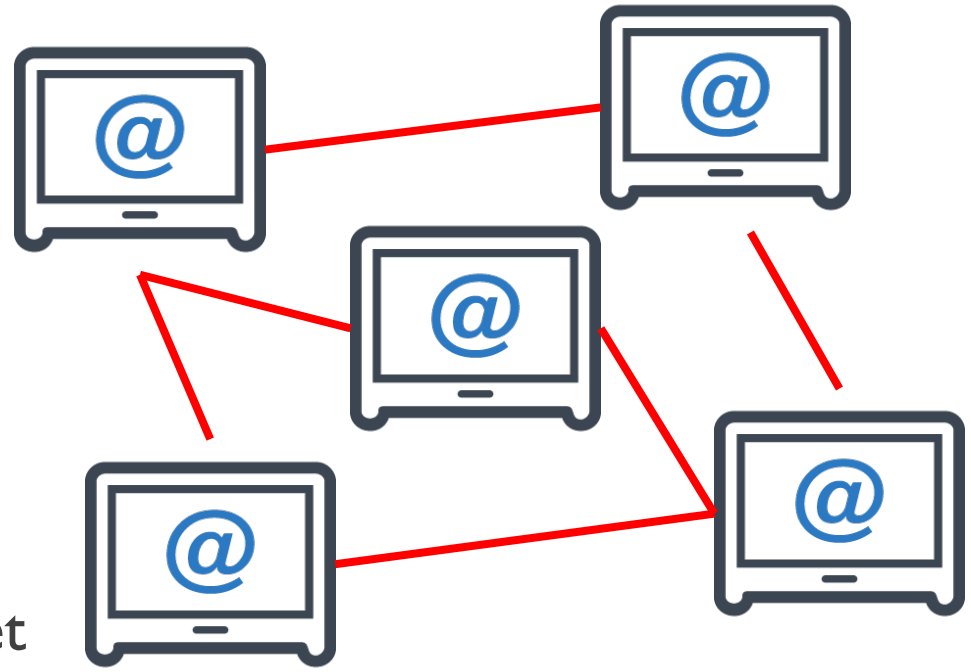
About me


70°
60°
50°
40°
EQUADOR
10°
20°
30°
TRÓPICO DE CAPRICÓRNIO
Boa Vista
RORAIMA
Serra Pacaraima
AMAPÁ
Macapá
Manaus
AMAZONAS
PARÁ
Belém
São Luís
Fortaleza
Fernando de Noronha (PE)
MARANHÃO
CEARÁ
RIO GRANDE DO NORTE
Teresina
Natal
PIAUÍ
PARAÍBA
João Pessoa
PERNAMBUCO
Recife
ACRE
Porto Velho
Rio Branco
Palmas
ALAGOAS
Maceió
RONDÔNIA
TOCANTINS
SERGIPE
Aracajú
MATO GROSSO
BAHIA
Salvador
Cuiabá
BRASÍLIA-DF
GOIAS
Goiânia
MINAS GERAIS
MATO GROSSO DO SUL
Belo Horizonte
ESPÍRITO SANTO
Campo Grande
Vitória
SÃO PAULO
RIO DE JANEIRO
São Paulo
Rio de Janeiro
PARANÁ
Curitiba
SANTA CATARINA
Florianópolis
RIO GRANDE DO SUL
Porto Alegre
Região Norte
Região Nordeste
Região Sudeste
Região Sul
Região Centro-Oeste
0
1160
Km
BRASIL - Político


Para entender o tamanho do Brasil
GUIANA
TAJIQUISTÃO
MONGÓLIA
ANGOLA
ITÁLIA
GRÉCIA
BOSNIA
CROÁCIA
PORTUGAL
HAITI
GUINÉ
NEPAL
NOVA ZELÂNDIA
LAOS
ISRAEL
FRANÇA
VENEZUELA
TRINIDAD E TOBAGO
FINLÂNDIA
ESPANHA
SUIÇA
ALEMANHA
REINO UNIDO
DINAMARCA
SENEGAL
HUNGRIA
EQUADOR
LÍNGUA GALEGO PORTUGUESA

From standalone to connected pieces of software

- Software components depend on one another
- Key differences from the past
  - Dynamic environment
  - Little can be assumed from the behaviour of other components
  - Intensive communication based on the internet

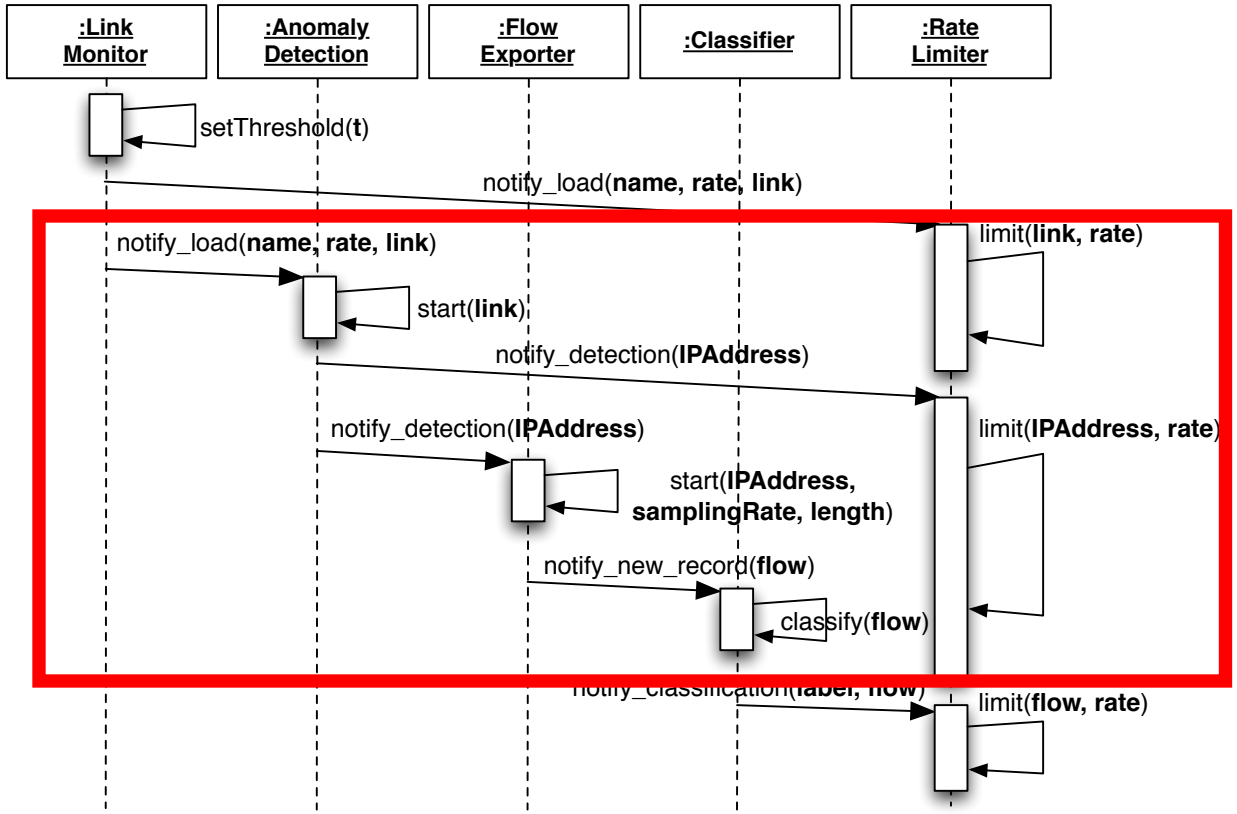


RESILIENCE and EFFICIENCY

Automated Management of Remediation Actions

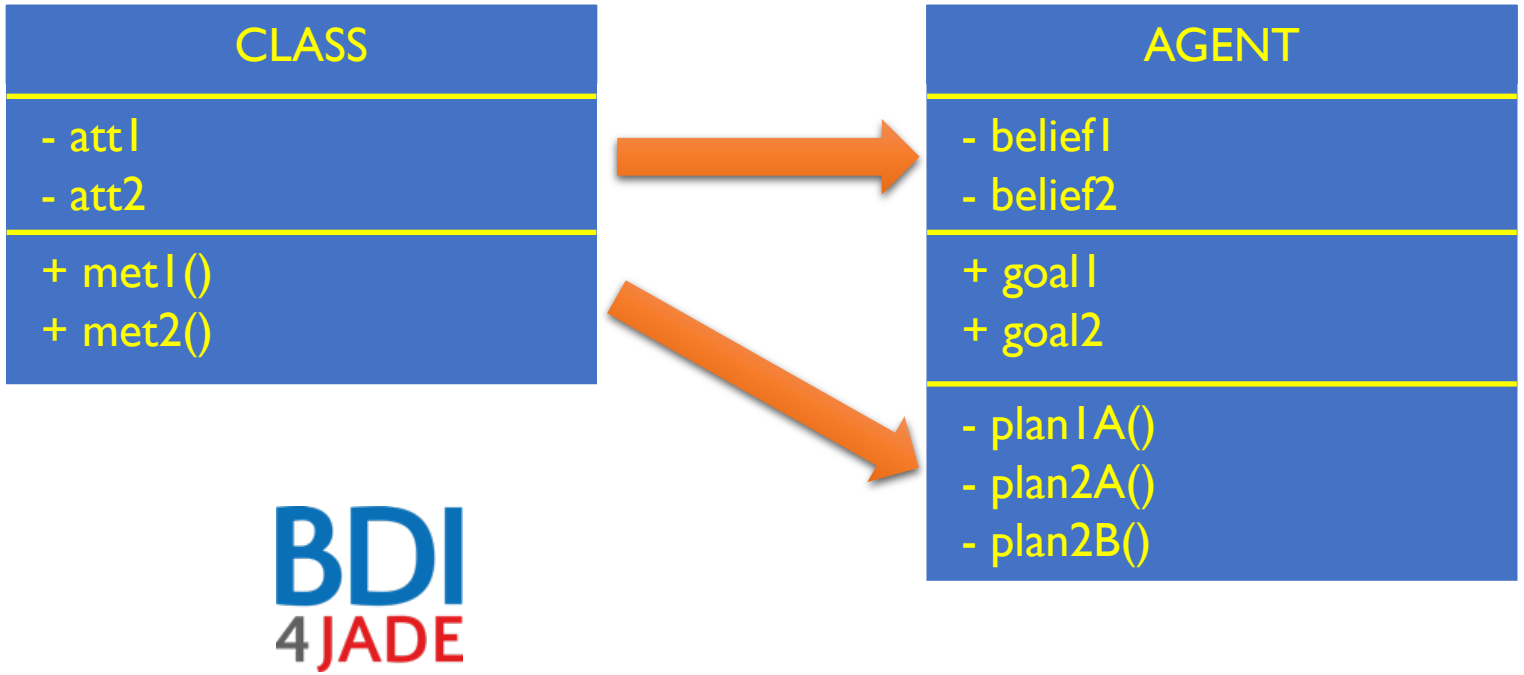
- Remediation Action
  - An action that mitigates the consequences/effects of a problem
- Why?
  - Causes of the problem are unknown
  - Addressing the causes takes too long

- Example

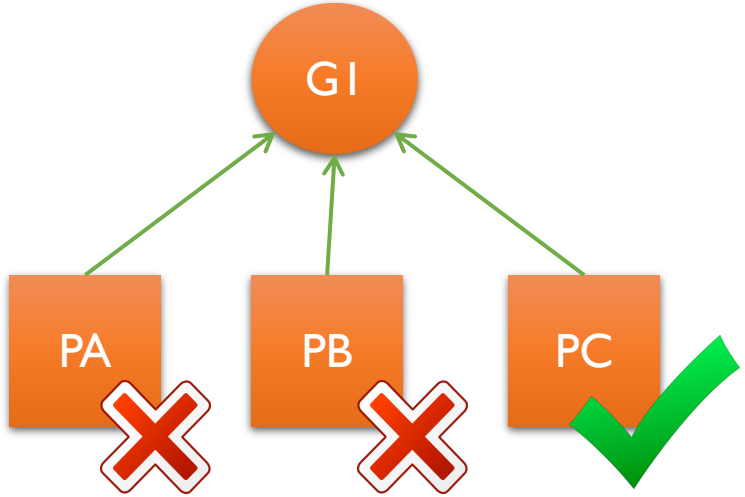


Automated Management of Remediation Actions

- Belief-Desire-Intention (BDI) Agents



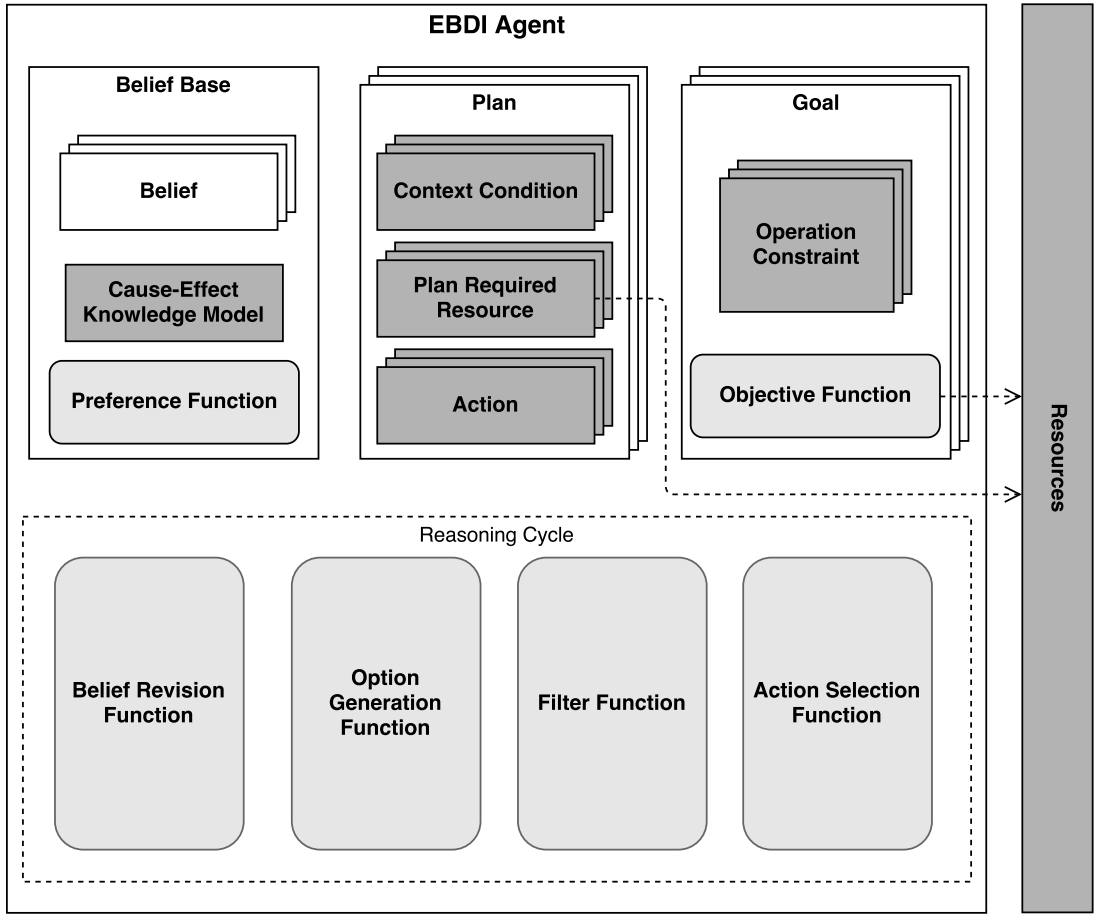
BDI 4JADE



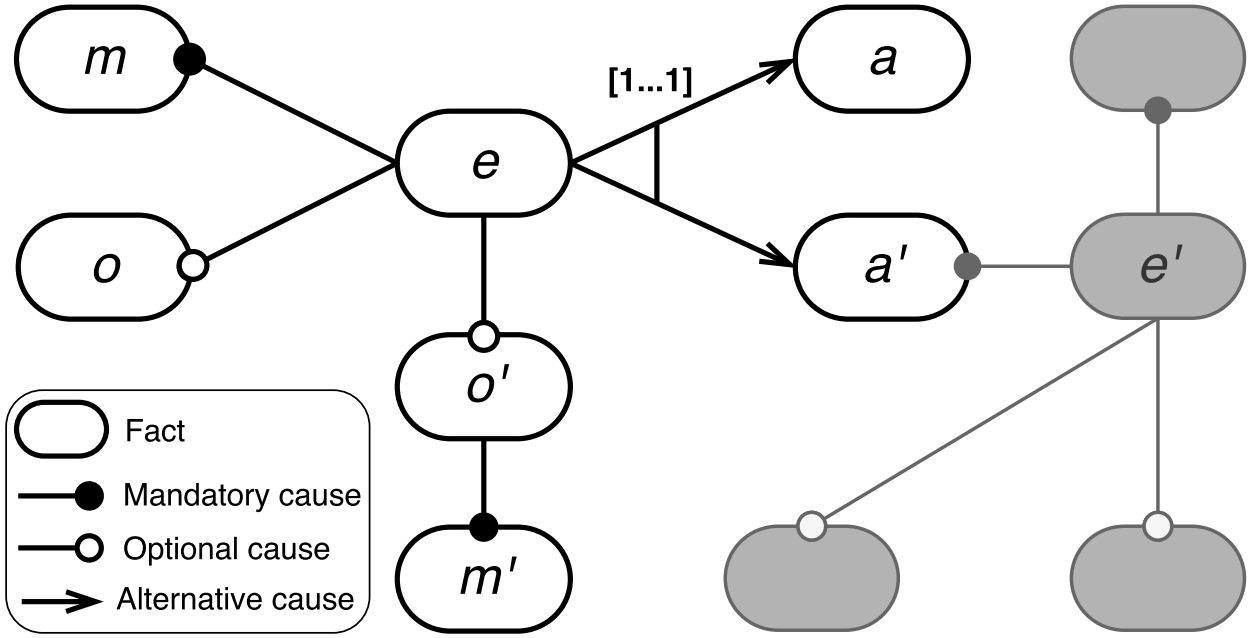
Reasoning cycle:
1. Update beliefs
2. Generate goals
3. Filter goals
4. Select plan

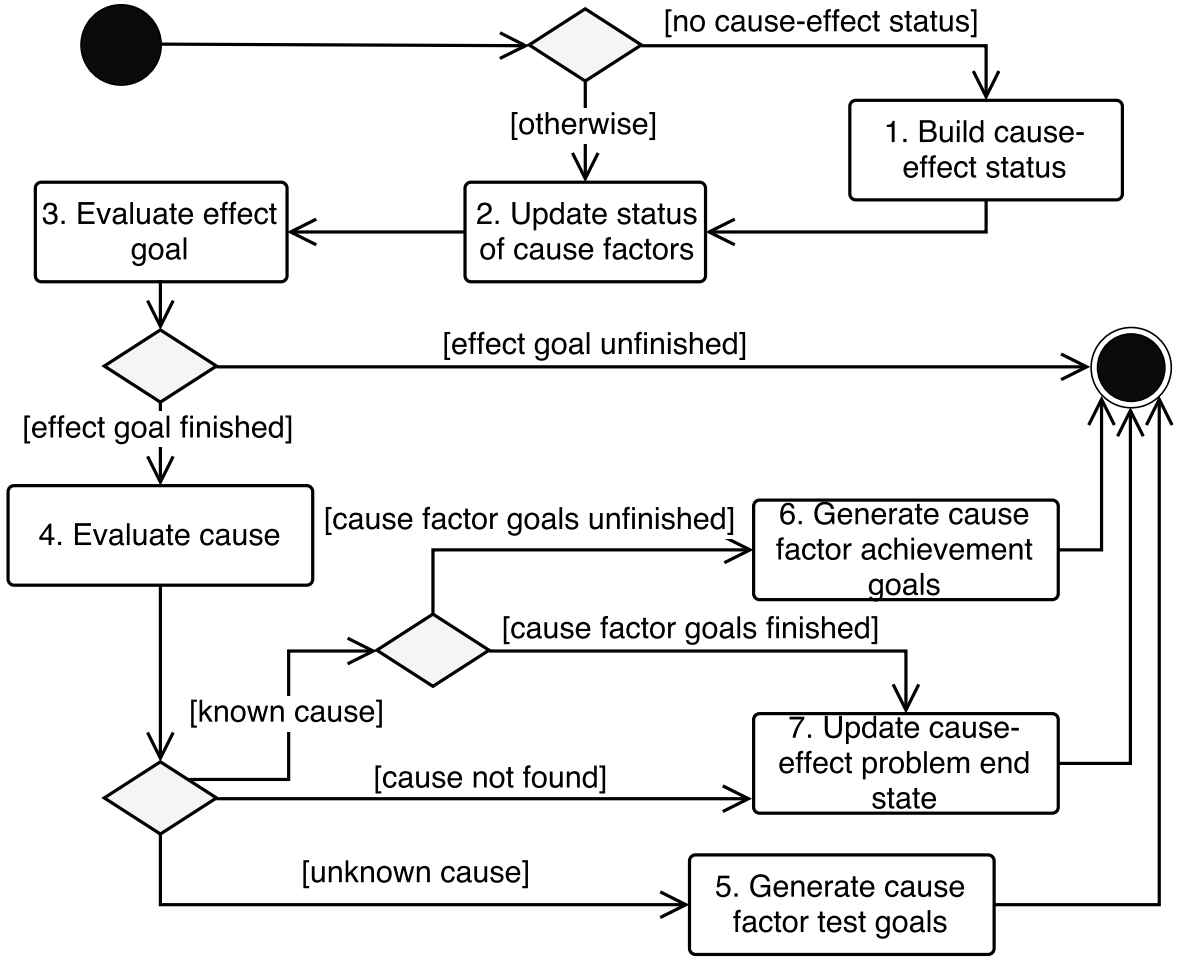
Automated Management of Remediation Actions

1. Constrained Goals/Plan Selection
   - Goal metadata to indicate restrictions of how goals can be achieved
     - constraints + utility-function
2. Cause-effect Knowledge Model
   - Used to identify possible problem causes
3. Goal Generation
   - Identify problem causes
   - Address identified causes



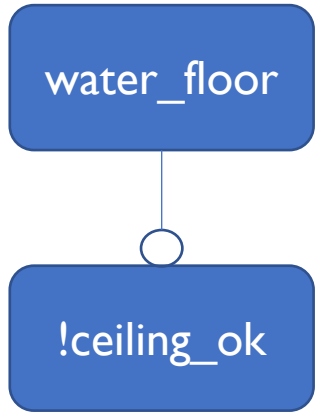
Automated Management of Remediation Actions



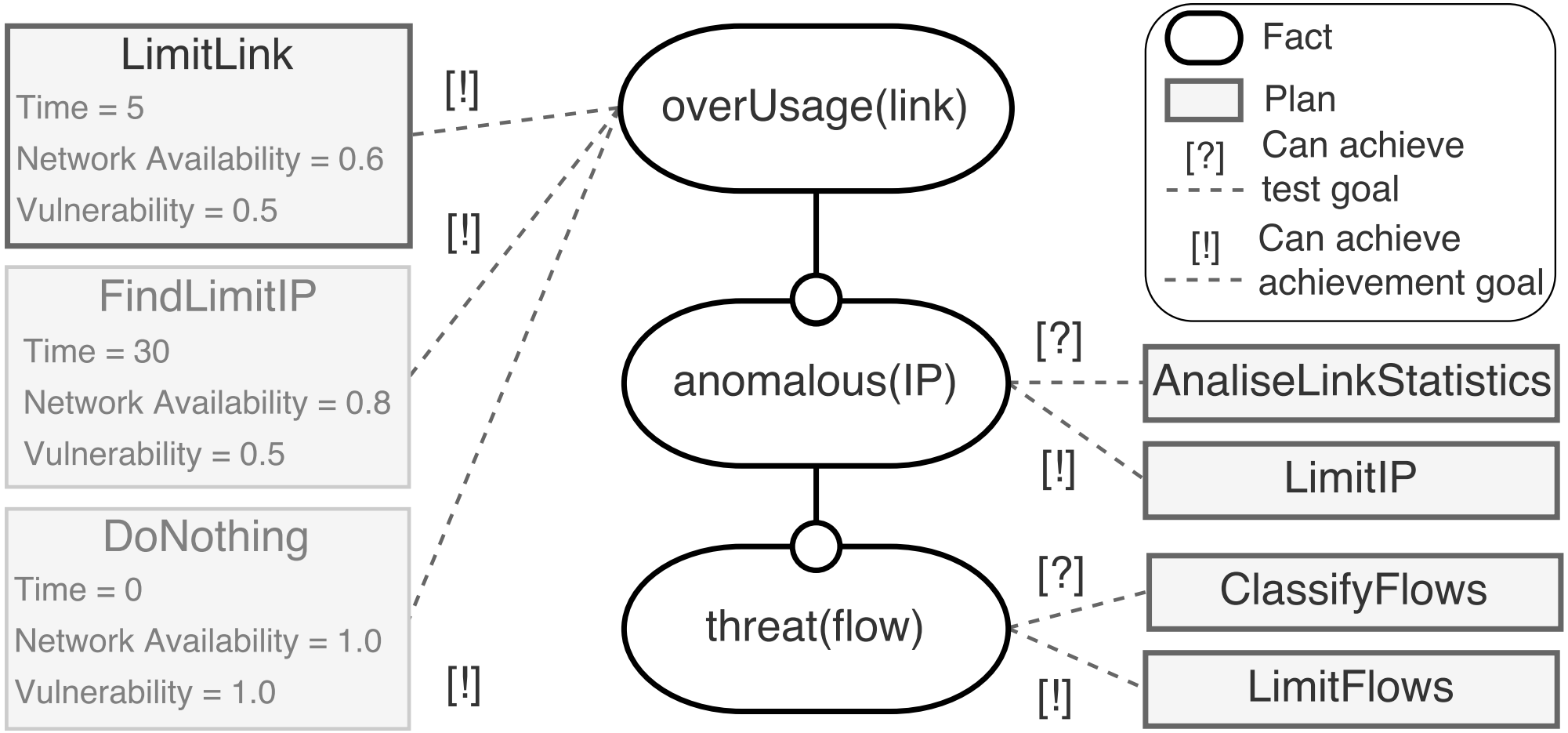


Automated Management of Remediation Actions

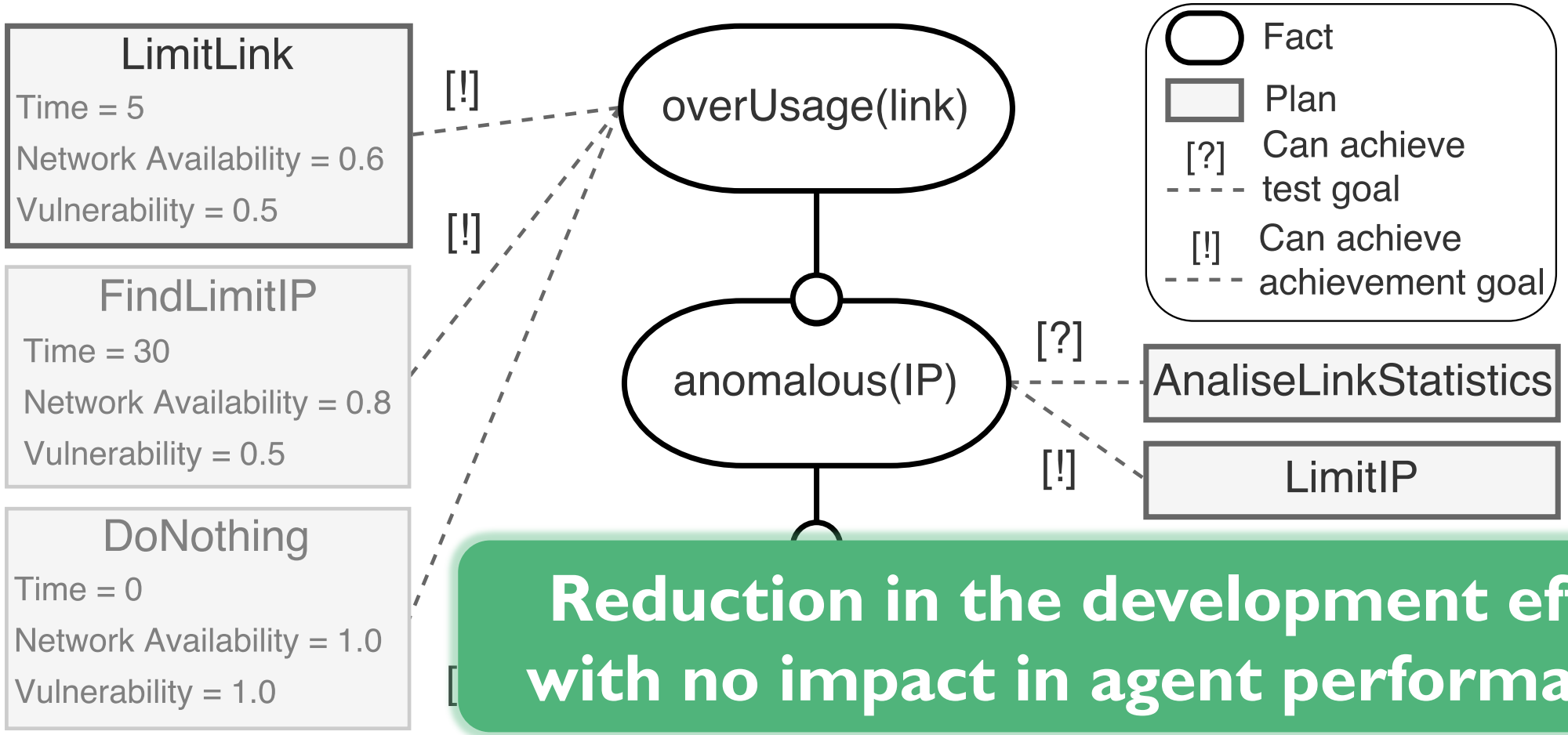
- Goal
  - ~water_floor
  - min time
- Plans
  - place_bucket
    - Post-condition: ~water_floor
    - Resources: 1min
  - repair_ceiling
    - Pre-condition: have_tools
    - Post-conditions:
      - ~water_floor
      - ceiling_ok
    - Resources: 60min



Automated Management of Remediation Actions

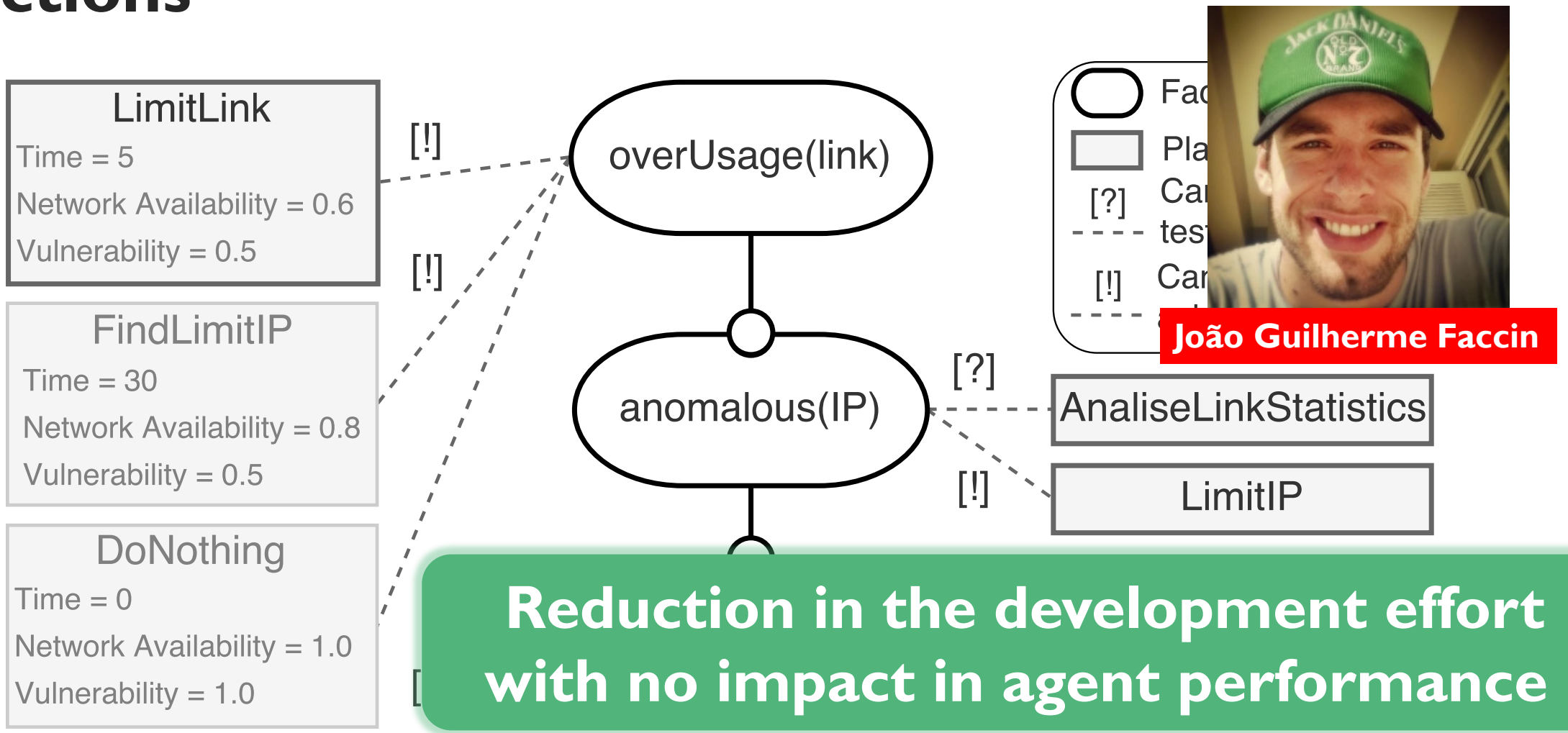


Automated Management of Remediation Actions



Reduction in the development effort with no impact in agent performance

Automated Management of Remediation Actions



Reduction in the development effort with no impact in agent performance

Application-level Caching

```
public class ProductService {
  /** some business logic */
  public List <Product> search(String query, boolean checkWarehouse) {
    products = ProductRepository.search(query);
    if (checkWarehouse)
      products.addAll(WarehouseAccess.searchProducts(query));
    return products;
 }
}
```

Application-level Caching

```java
public class ProductService {
  /** some business logic */
  public List <Product> search(String query, boolean checkWarehouse){
    Cache cache = Cache.getInstance("productsCache");
    Cache cacheKey = "products:" + query; //key definition
    synchronized(this) {
     List <Product> products = (List <Product>)cache.get(cacheKey);
     if (products == null){
      products = ProductRepository.search(query);
      if (checkWarehouse)
       products.addAll(WarehouseAccess.searchProducts(query));
      cache.put(cacheKey, products, 30); //TTL 30 seconds
     }
    }
    return products;
  }
}
```

Frequently retrieved or expensive to compute

Application-level Caching

```
public class ProductService {
 /** some business logic */
 public List <Product> search(String query, boolean
  Cache cache = Cache.getInstance("productsCache");
  Cache cacheKey = "products:" + query; //key defin
  synchronized(this) {
   List <Product> products = (List <Product>)cache.get(cacheKey);
   if (products == null){
    products = ProductRepository.search(query);
    if (checkWarehouse)
     products.addAll(WarehouseAccess.searchProducts(query));
    cache.put(cacheKey, products, 30); //TTL 30 seconds
   }
  }
  return products;
 }
}
```

Demands extensive knowledge

Application-level Caching

```
public class ProductService {
 /** some business logic */
 public List <Product> search(String query, boolean checkWarehouse){
  Cache cache = Cache.getInstance("productsCache");
  Cache cacheKey = "products:" + query; //key definition
  synchronized(this) {
   List <Product> products = (List <Product>)cache.get(cacheKey);
   if (products == null){
    products = ProductRepository.search(query);
    if (checkWarehouse)
     products.addAll(WarehouseAccess.searchProducts(query));
    cache.put(cacheKey, products, 30); //TTL 30 seconds
   }
  }
  return products;
 }
}
```

.java

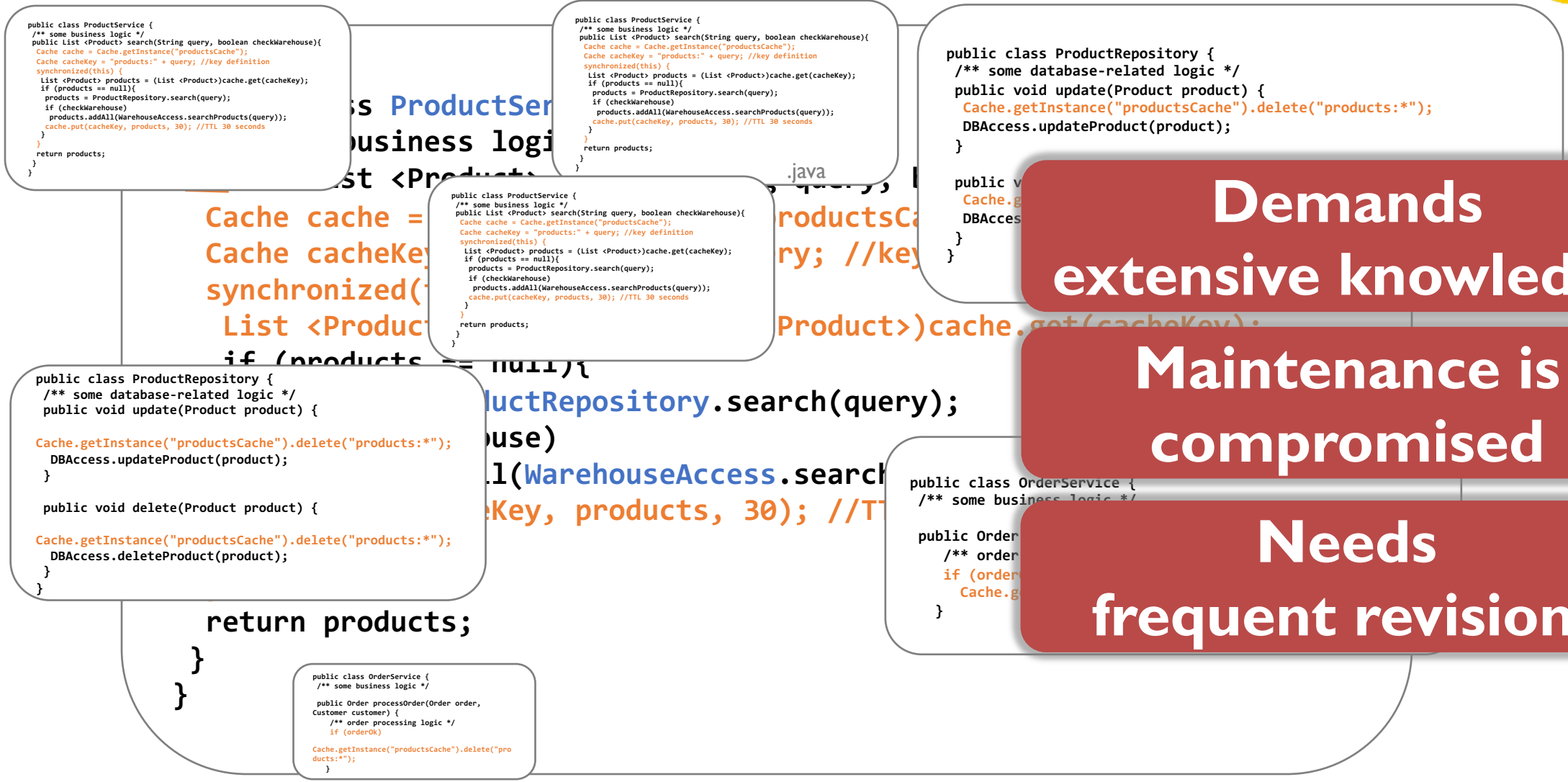
```
public class ProductRepository {
 /** some database-related logic */
 public void update(Product product) {
  Cache.getInstance("productsCache").delete("products:*");
  DBAccess.updateProduct(product);
 }

 public void delete(Product product) {

Cache.getInstance("productsCache").delete("products:*");
  DBAccess.deleteProduct(product);
 }
}
```

```
public class OrderService {
 /** some business logic */

 public Order processOrder(Order order,
Customer customer) {
    /** order processing logic */
    if (orderOk)

Cache.getInstance("productsCache").delete("pro
ducts:*");
   }
```

Demands extensive knowledge

Maintenance is compromised

Application-level Caching



Demands extensive knowledge

Maintenance is compromised

Needs frequent revisions

Application-level Caching

Understanding

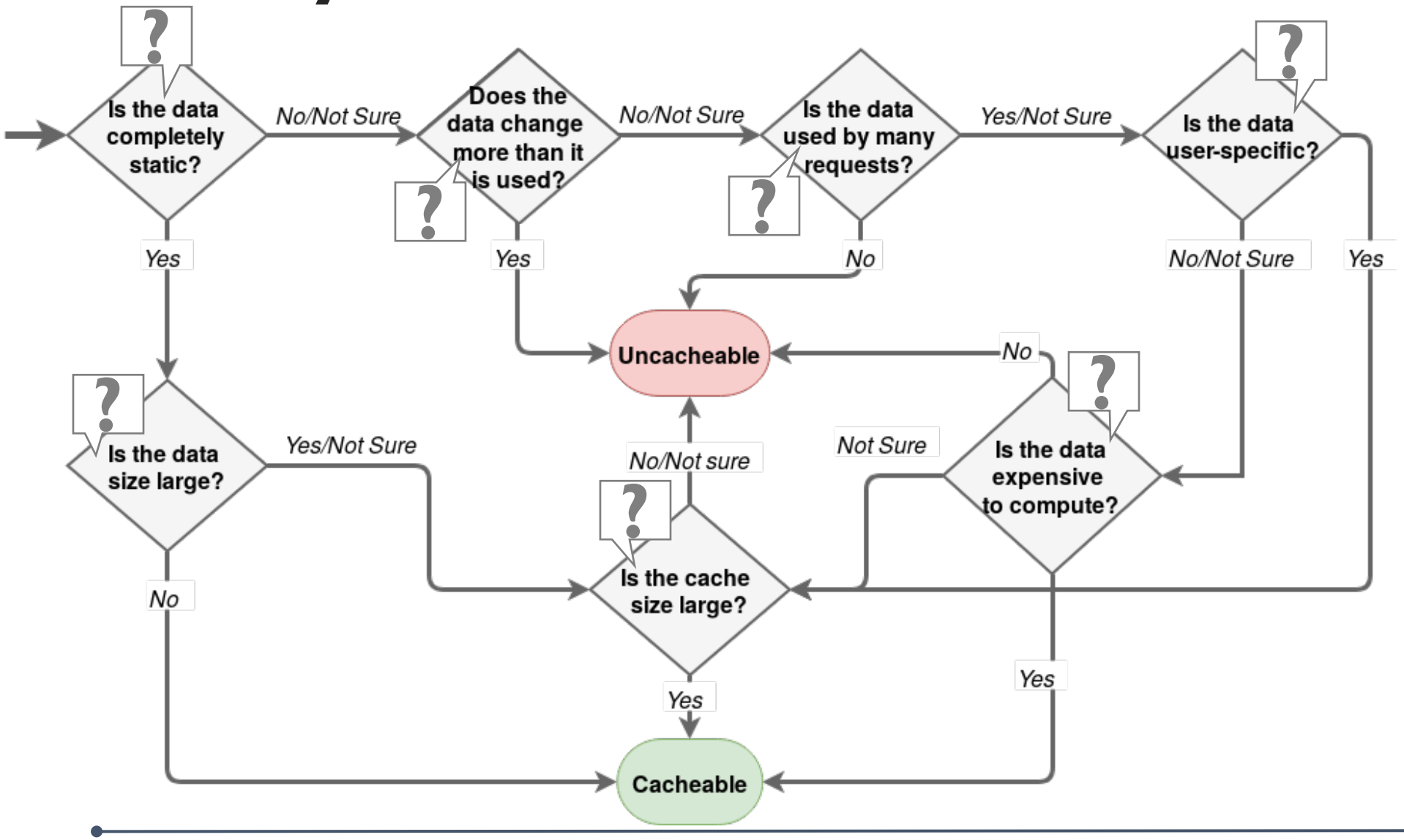
- Survey and Taxonomy of existing approaches
- Analysis of existing practices

Support

- Guidelines and Patterns
- Approach and framework to automate content selection

MERTZ, J. ; NUNES, I. . Understanding Application-Level Caching in Web Applications: A Comprehensive Introduction and Survey of State-of-the-Art Approaches. ACM COMPUTING SURVEYS, v. 50, p. 1-34, 2017.

Cacheability Pattern



Cacheability Pattern

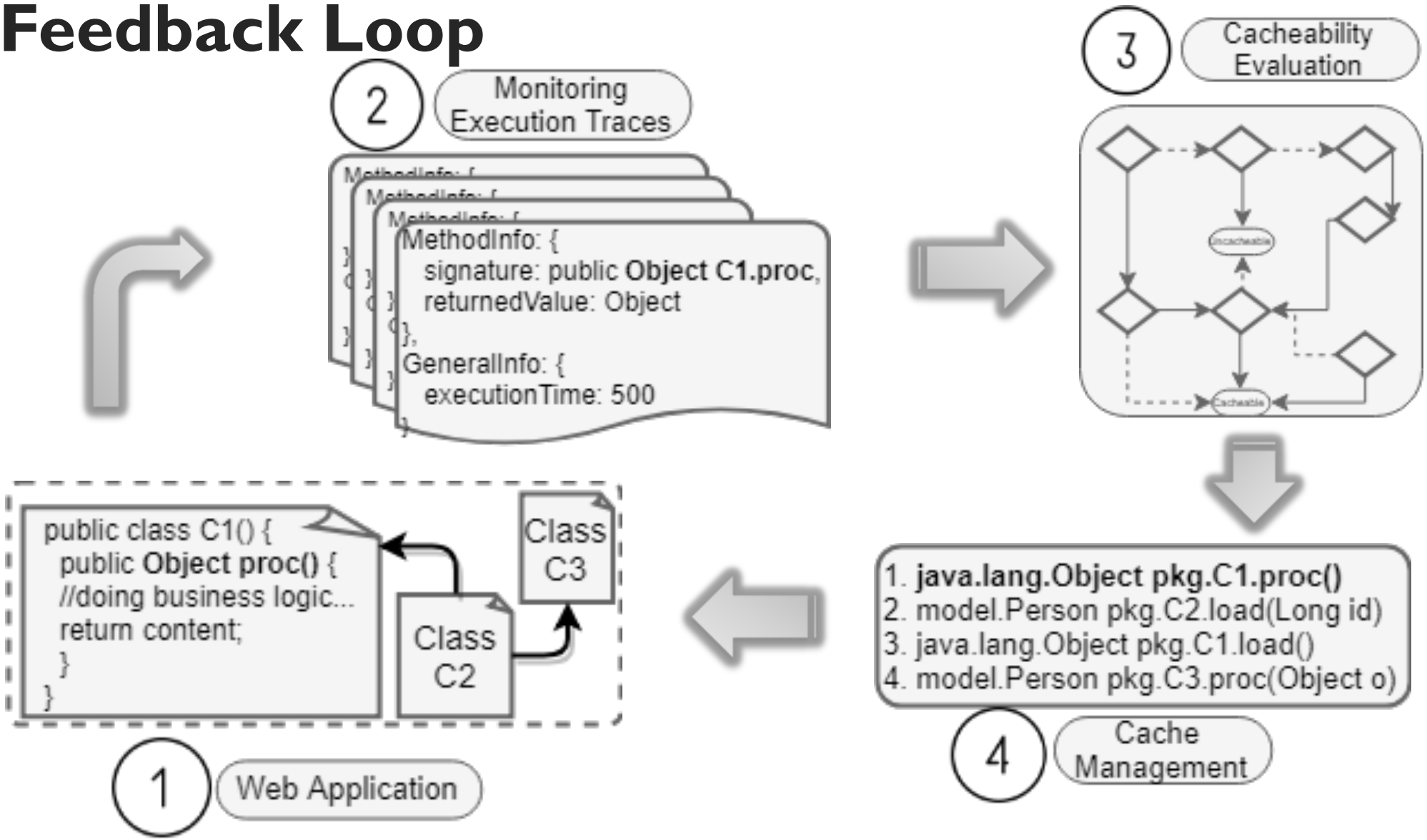
percentage of calls with the same parameters returning different values

$$changeability(m) > \mu_{ch} + k \times \sigma_{ch}$$

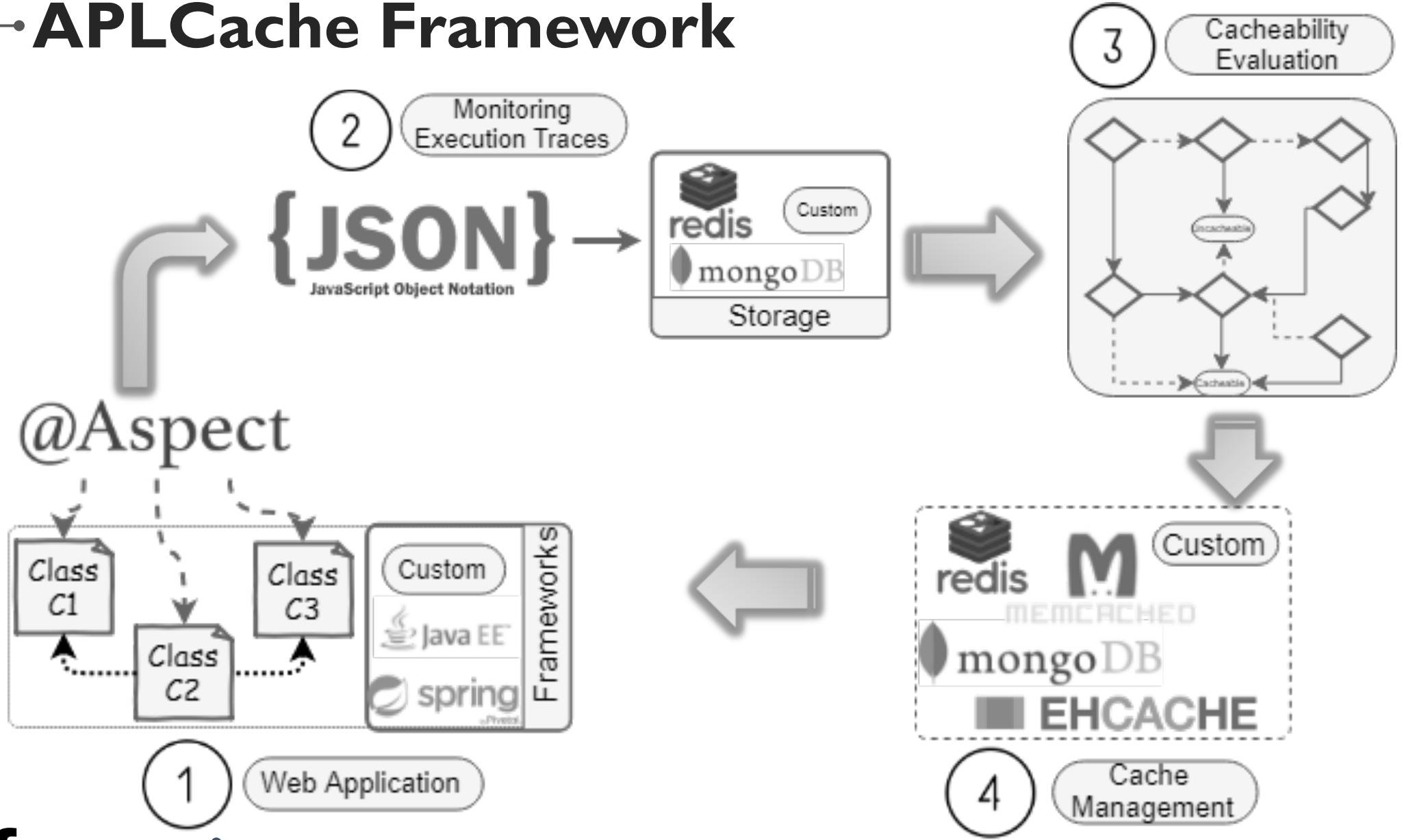
Yes

k standard deviations above the mean of this metric considering all methods

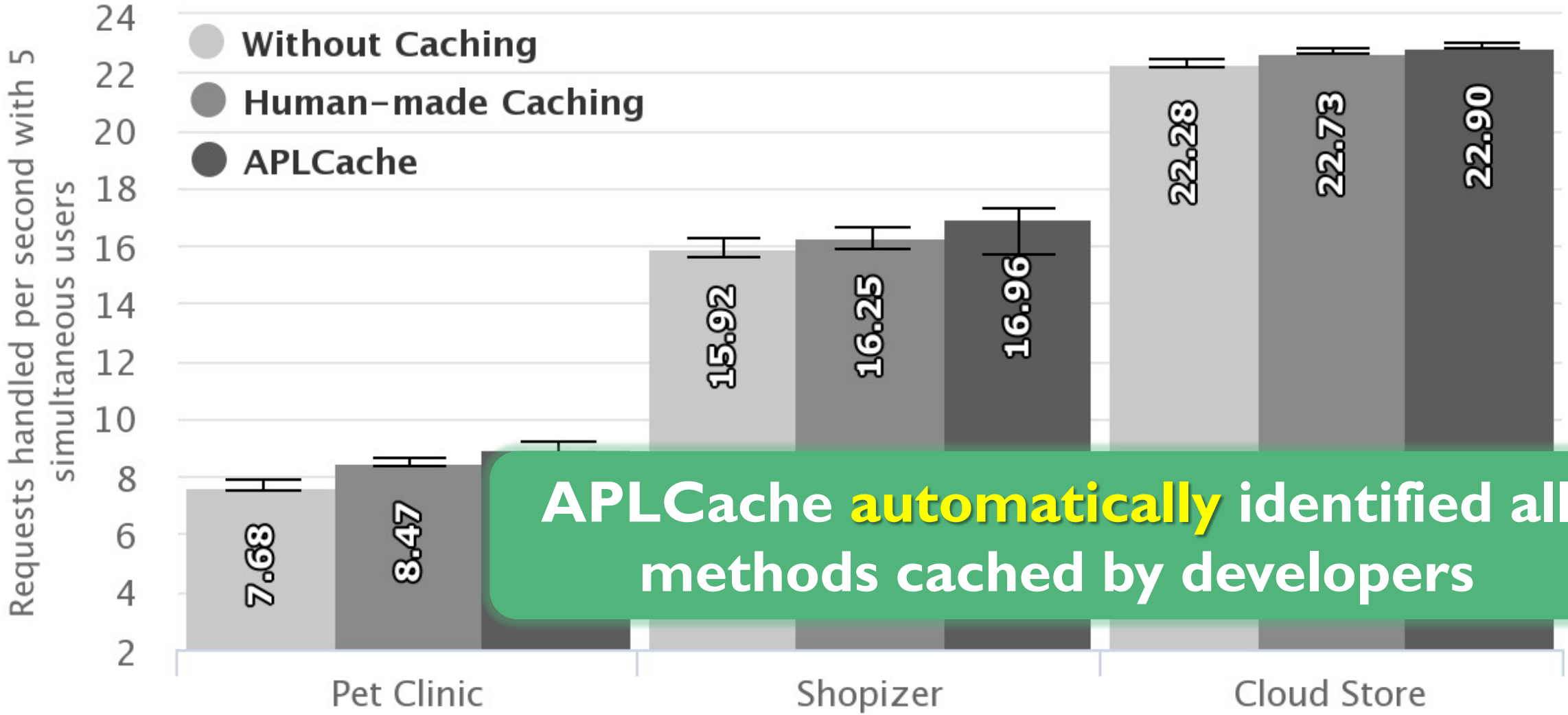
Feedback Loop



APLCache Framework



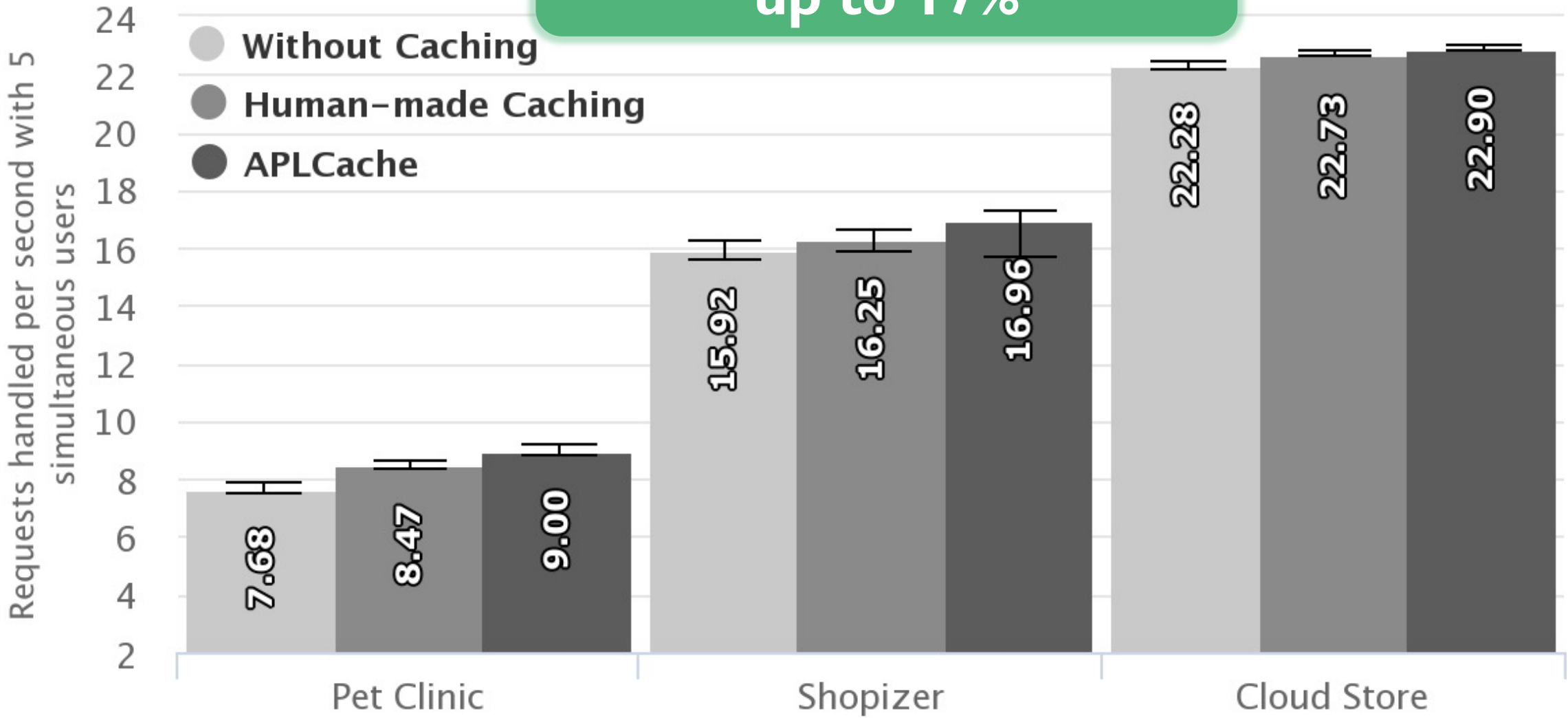
Evaluation



APLCache automatically identified all methods cached by developers

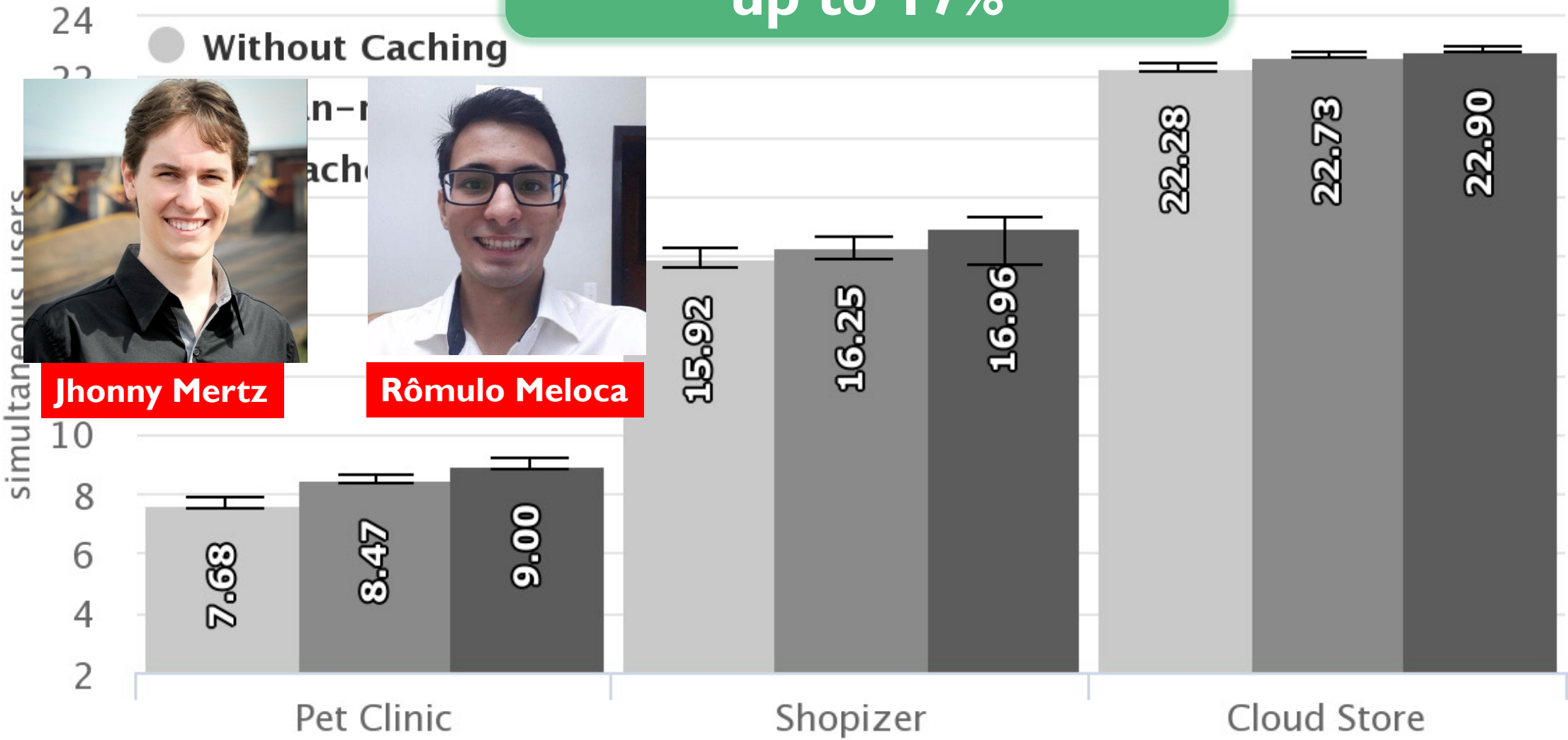
Evaluation

Increased throughput up to 17%



Evaluation

Increased throughput up to 17%



Jhonny Mertz

Rômulo Meloca

Conclusion

- Automated management of remediation actions
  - Promotes resilience
- Application-level caching
  - Can be used for improving resilience in case of, e.g., communication failure
  - Promotes efficiency, avoiding unnecessary calls
- What about IoT?
  - Current application domain
    - Mitigation of CO leaking in Smart Homes
    - Use of caching in the iCasa project
      - with University of Grenoble

Key References

- Remediation Actions
  - FACCIN, J. ; NUNES, I. . Remediating critical cause-effect situations with an extended BDI architecture. EXPERT SYSTEMS WITH APPLICATIONS, v. 95, p. 190-200, 2018.
  - FACCIN, J. ; NUNES, I. . Cleaning Up the Mess: a Formal Framework for Autonomously Reverting BDI Agent Actions. SEAMS 2018.
- Application-level Caching
  - MERTZ, J. ; NUNES, I. . Automation of application-level caching in a seamless way. SOFTWARE-PRACTICE & EXPERIENCE, v. 48, p. 1218-1237, 2018.
  - MERTZ, J.; NUNES, I. A Qualitative Study of Application-level Caching. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, v. 43, p. 798-816, 2017.

http://inf.ufrgs.br/~ingridnunes/

.inf INSTITUTO DE INFORMÁTICA UFRGS

Prosoft

UFRGS UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL