CLOUD COMPUTING 2022
The Thirteenth International Conference on Cloud Computing, GRIDs, and Virtualization April 24, 2022 to April 28, 2022 - Barcelona, Spain (and online)
https://www.iaria.org/conferences2022/CLOUDCOMPUTING22.html

Metamodel and Patterns for Cloud Security and Privacy

Hironori Washizaki

Professor at Waseda University, Tokyo, Japan

washizaki@waseda.jp

WASEDA University

https://www.waseda.jp/culture/news/2020/04/30/10381/

Prof. Dr. Hironori Washizaki

- Professor and the Associate Dean of the Research Promotion Division at Waseda University in Tokyo
- Visiting Professor at the National Institute of Informatics
- Outside Directors of SYSTEM INFORMATION and eXmotion
- Research and education projects
  - Leading a large-scale grant at MEXT enPiT-Pro Smart SE
  - Leading framework team of JST MIRAI eAI project
- Professional contributions
  - **IARIA Fellow**
  - IEEE Computer Society Vice President for Professional and Educational Activities
  - Editorial Board Member of MDPI Education Sciences
  - Steering Committee Member of the IEEE Conference on Software Engineering Education and Training (CSEE&T)
  - Associate Editor of IEEE Transactions on Emerging Topics in Computing
  - Advisory Committee Member of the IEEE-CS COMPSAC
  - Steering Committee Member of Asia-Pacific Software Engineering Conference (APSEC)
  - Convener of ISO/IEC/JTC1 SC7/WG20

Metamodel and Patterns for Cloud Security and Privacy

Hironori Washizaki

Professor at Waseda University, Tokyo, Japan

washizaki@waseda.jp http://www.washi.cs.waseda.ac.jp/

- Tian Xia, Hironori Washizaki, Yoshiaki Fukazawa, Haruhiko Kaiya, Shinpei Ogata, Eduardo B. Fernandez, Takehisa Kato, Hideyuki Kanuka, Takao Okubo, Nobukazu Yoshioka and Atsuo Hazeyama, “CSPM: Metamodel for Handling Security and Privacy Knowledge in Cloud Service Development,” International Journal of Systems and Software Security and Protection (IJSSSP), Vol. 12, No. 2, IGI-Global, pp.1-18, 2021.
- Hironori Washizaki, Tian Xia, Natsumi Kamata, Yoshiaki Fukazawa, Hideyuki Kanuka, Takehisa Kato, Masayuki Yoshino, Takao Okubo, Shinpei Ogata, Haruhiko Kaiya, Atsuo Hazeyama, Takafumi Tanaka, Nobukazu Yoshioka, G Priyalakshmi, “Systematic Literature Review of Security Pattern Research,” Information, Vol. 12, No. 1:36, MDPI, pp.1-27, 2021.
- Eduardo B. Fernandez, Nobukazu Yoshioka, Hironori Washizaki, Madiha H. Syed, “Modeling and Security in Cloud Ecosystems,” Future Internet, Special Issue Security in Cloud Computing and Big Data, Vol.8, No.13(2), pp.1-15, 2016.
- Tian Xia, Hironori Washizaki, Takehisa Kato, Haruhiko Kaiya, Shinpei Ogata, Eduardo B. Fernandez, Hideyuki Kanuka, Masayuki Yoshino, Dan Yamamoto, Takao Okubo, Nobukazu Yoshioka and Atsuo Hazeyama, “Cloud Security and Privacy Metamodel: Metamodel for Security and Privacy Knowledge in Cloud Services,” 6th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2018), short paper, pp.379-386, FUNCHAL, MADEIRA – Portugal 22 – 24 January, 2018.

Agenda

- Paradigm shifts in new software engineering
- Pattern language
- Security patterns
- Metamodel and Patterns for Cloud Security and Privacy

What is software engineering?

- "Application of systematic, disciplined, quantifiable approach to development, operation, and maintenance of software" – SWEBOK 2014
- Guide to the Software Engineering Body of Knowledge (SWEBOK)

- Software Requirements
- Software Design
- Software Construction
- Software Testing

- Software Maintenance
- Software Configuration Management
- Software Engineering Management
- Software Engineering Process

- Software Engineering Tools and Methods
- Software Quality
- Software Engineering Professional Practice
- Software Engineering Economics

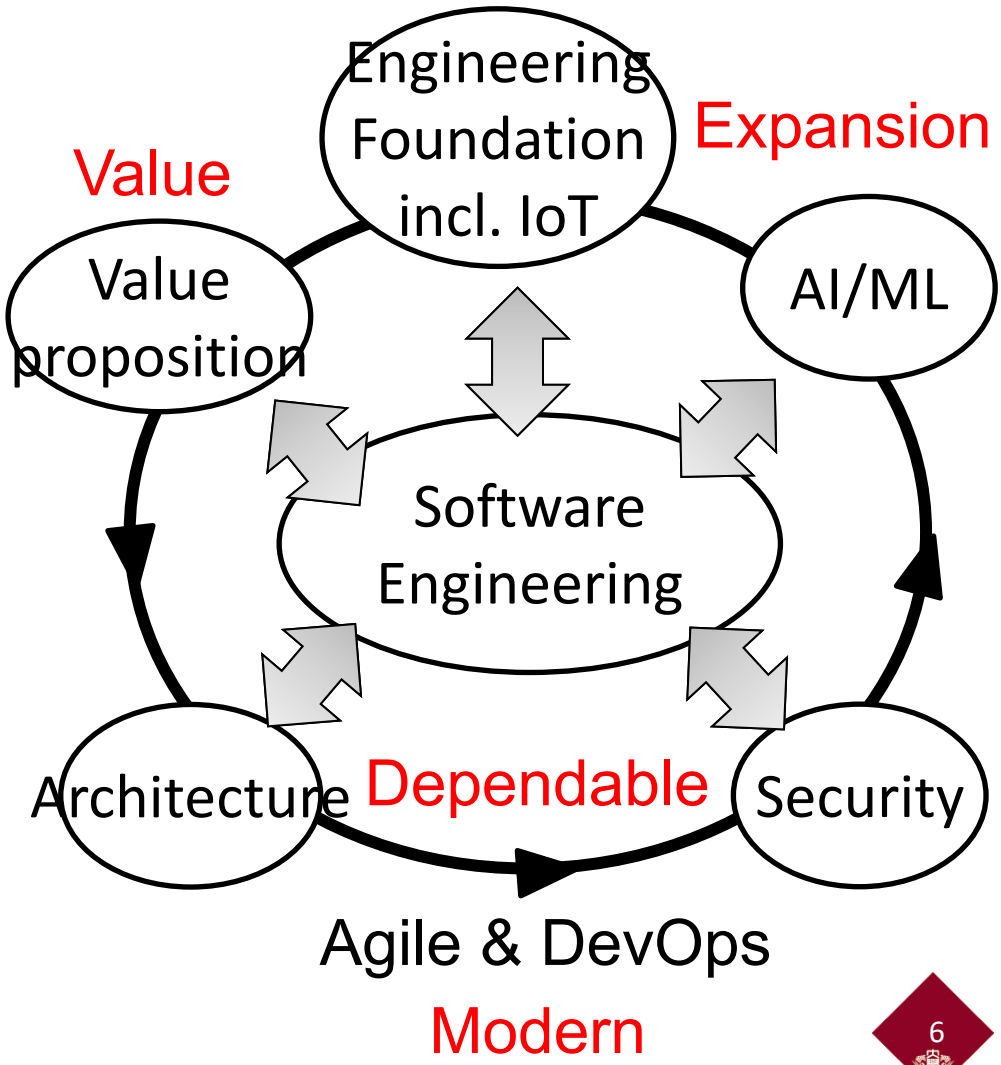
- Computing Foundations
- Mathematical Foundations
- Engineering Foundations

Vision of SWEBOK 2022 (subject to change)

(Evolution lead: Hironori Washizaki, since 2018-)

https://www.computer.org/volunteering/boards-and-committees/professional-educational-activities/software-engineering-committee/swebok-evolution

- Expansion of SE
  - AI/Machine Learning Engineering
  - Restructuring foundation areas incl. Internet of Things (IoT)
- Value in SE
  - Value proposition
- Dependable SE
  - Architecture
  - Security
- Modern SE
  - Agile
  - DevOps



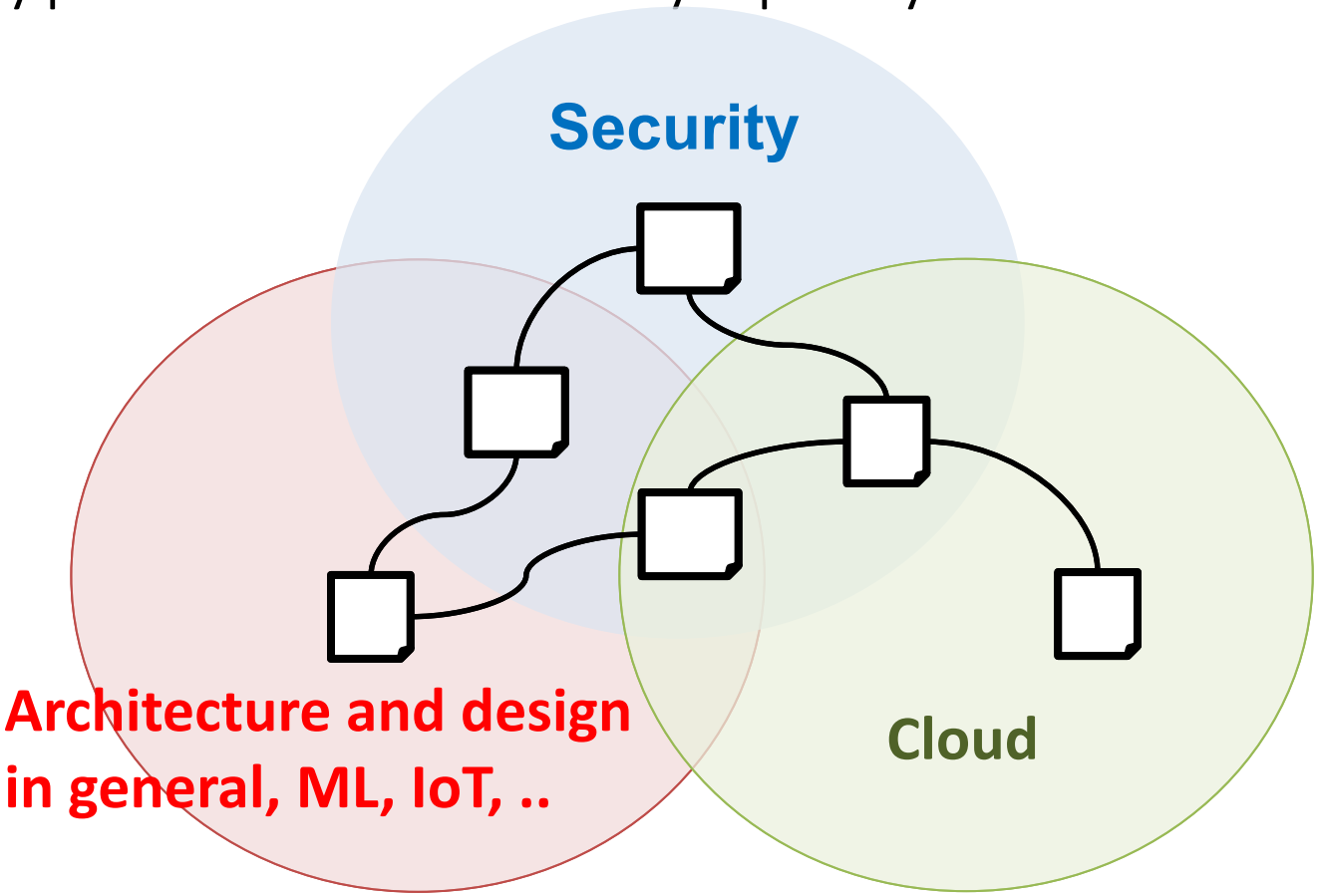
Paradigm shifts in "new" software engineering

	Current	New
Scope and perspective	Software systems	Software systems, **business, society and related disciplines**
Process	Planned, static, common, and closed	**Adaptive, dynamic, diverse, and open**
Focus	Specification	**Value, data, and speed**
Thinking	Cognitive (logical) or affective (design)	Cognitive (logical), affective (design), and **conative (conceptual)**
Inference	Deduction and analogy	Deduction, analogy, **induction, and abduction**

Hironori Washizaki, Junzo Hagimoto, Kazuo Hamai, Mitsunori Seki, Takeshi Inoue, Shinya Taniguchi, Hiroshi Kobayashi, Kenji Hiranabe and Eiichi Hanyuda, "Framework and Value-Driven Process of Software Engineering for Business and Society (SE4BS)," 5th International Conference on Enterprise Architecture and Information Systems (EAIS 2020)

Problem and goal

- Cloud computing is one of the key enablers of digital transformations.
- Security must be a critical cross-cutting concern in cloud and any other software.
- We are conducting systematic literature reviews to reveal landscapes of security patterns and cloud security & privacy.



Agenda

- Paradigm shifts in new software engineering
- Pattern language
- Security patterns
- Metamodel and Patterns for Cloud Security and Privacy

Street Cafe

Problem: Needs to have a place where people can sit lazily, legitimately, be on view, and watch the world go by…

Solution: Encourage local cafes to spring up in each neighborhood. Make them intimate places, with several rooms, open to a busy path …

Alexander, Christopher, et al. *A Pattern Language*. Oxford University Press, 1977.

https://unsplash.com/photos/8lKf54pc3qk

https://unsplash.com/photos/zACLEreWKXE

Towards a pattern language

… OK, so, to attract many people to our city, **Small Public Square**s should be located in the center. At the **Small Public Square**, make **Street Cafe**s be **Opening to the Street** …

https://unsplash.com/photos/EdpbTj3Br-Y

Small Public Square

https://unsplash.com/photos/zFoRwZirFvY

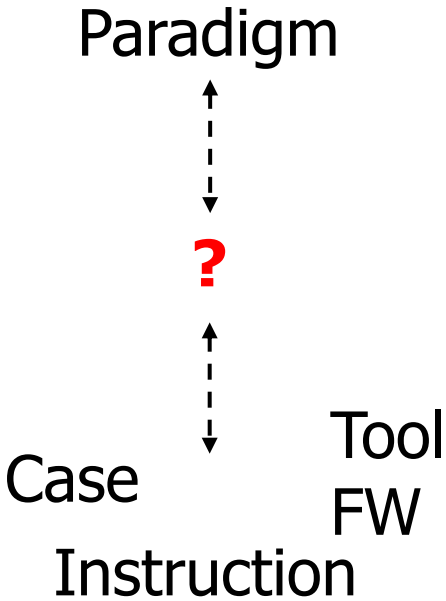
Street Cafe

https://unsplash.com/photos/GqurqYbj7aU

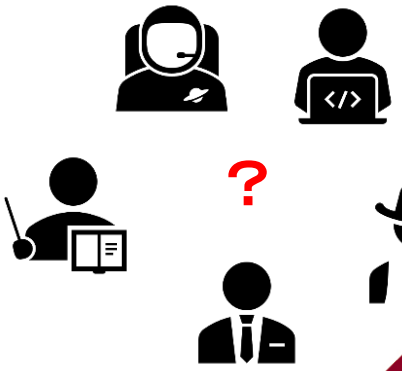
Opening to the Street

New SE needs pattern (language)!

- **Bridge** between abstract paradigms and concrete cases/tools
  - Verbalizing and documenting Know-Why (context), What (problem) and How (solution)
  - Reusing solutions and problems
  - Getting consistent architecture



- **Common language** among stakeholders
  - Security engineers, software engineers, hardware engineers, network engineers, domain experts, data analysist, …

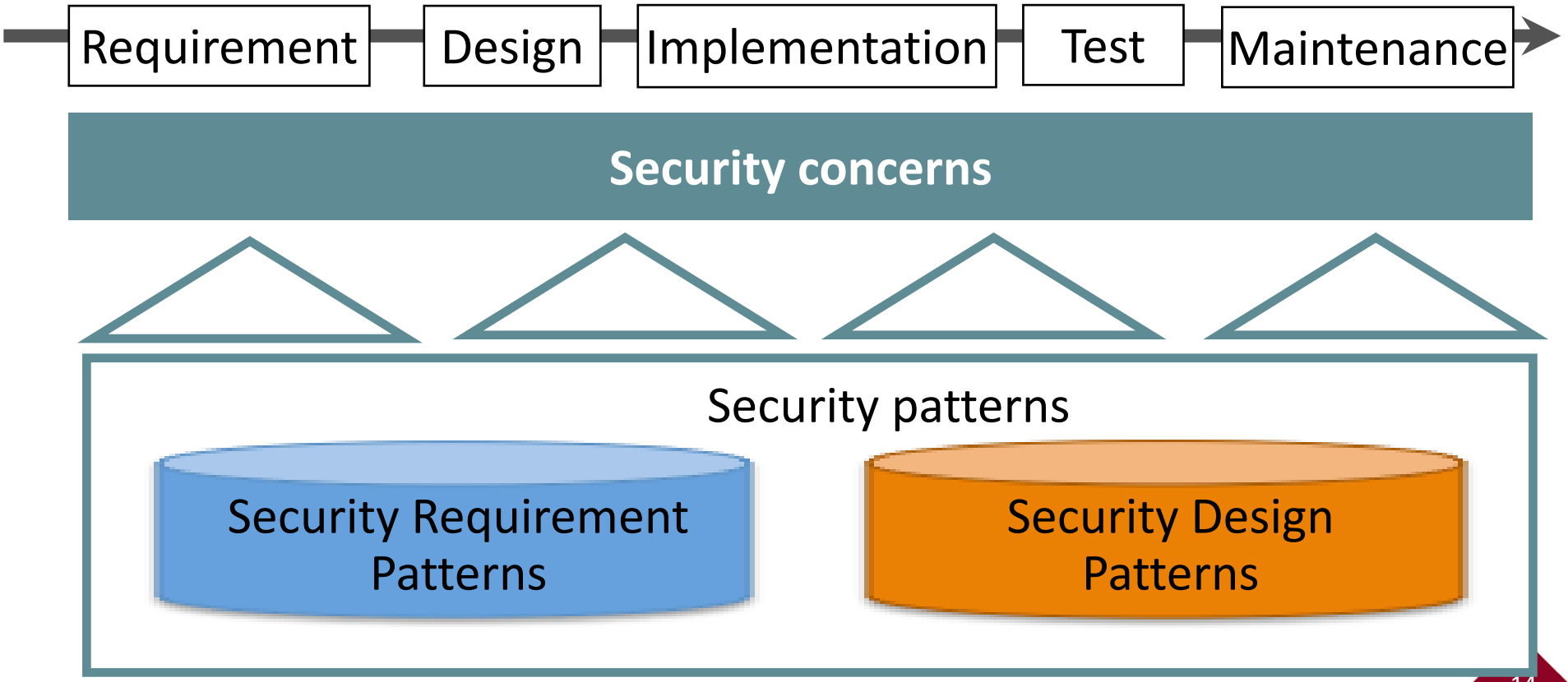


Agenda

- Paradigm shifts in new software engineering
- Pattern language
- Security patterns
- Metamodel and Patterns for Cloud Security and Privacy

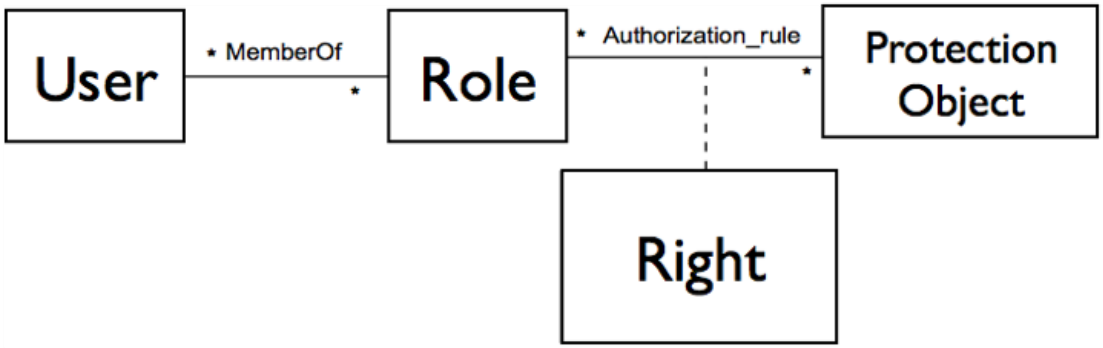
Security concerns must be addressed at any phase

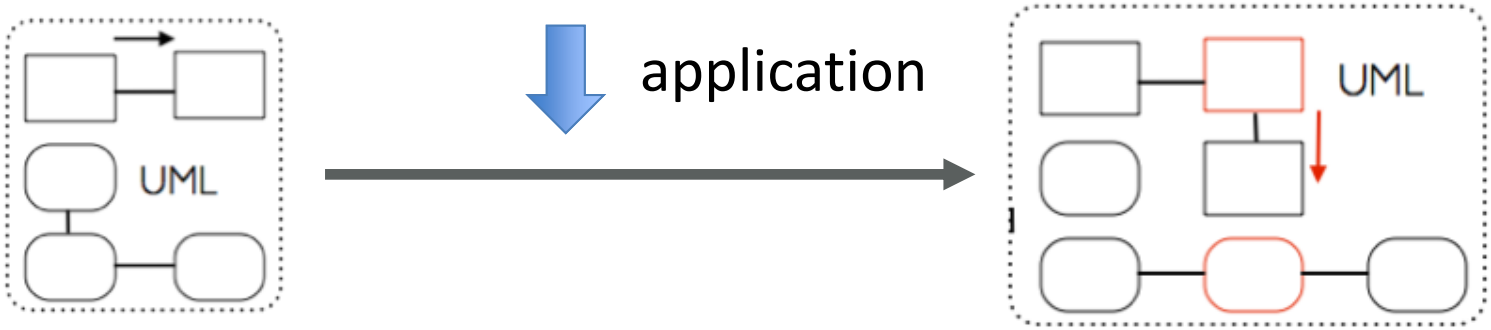
- Patterns are recurrent problems and solutions under specific contexts from requirements to maintenance



Example of security pattern

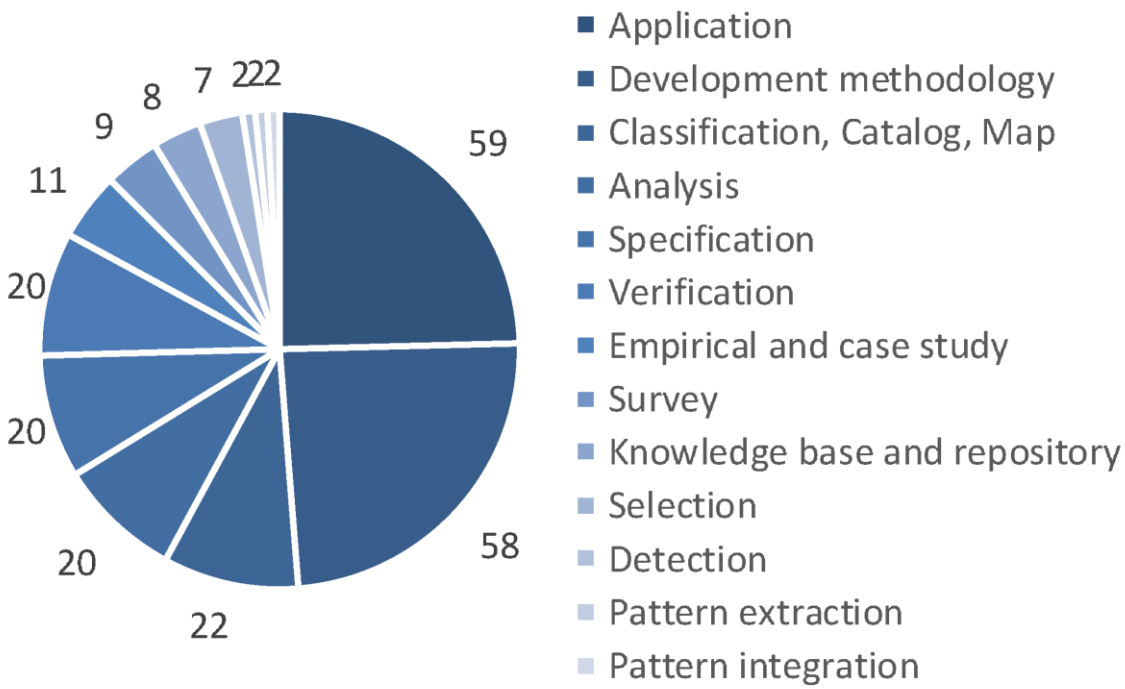
- Name: ***Role-based access control (RBAC)***
- Problem: How do we assign rights to people based on their functions or tasks?
- Solution: Assign users to roles and give rights to these roles so they can perform their tasks.
- Related patterns: ***Authorization,*** . . .





Systematic Literature Review of Security Pattern Research

- We categorize and analyze 240 papers to clarify state-of-the-art and future directions of security pattern research in terms of 13 facets including topics and security characteristics.
- E.g., breakdown of research topics



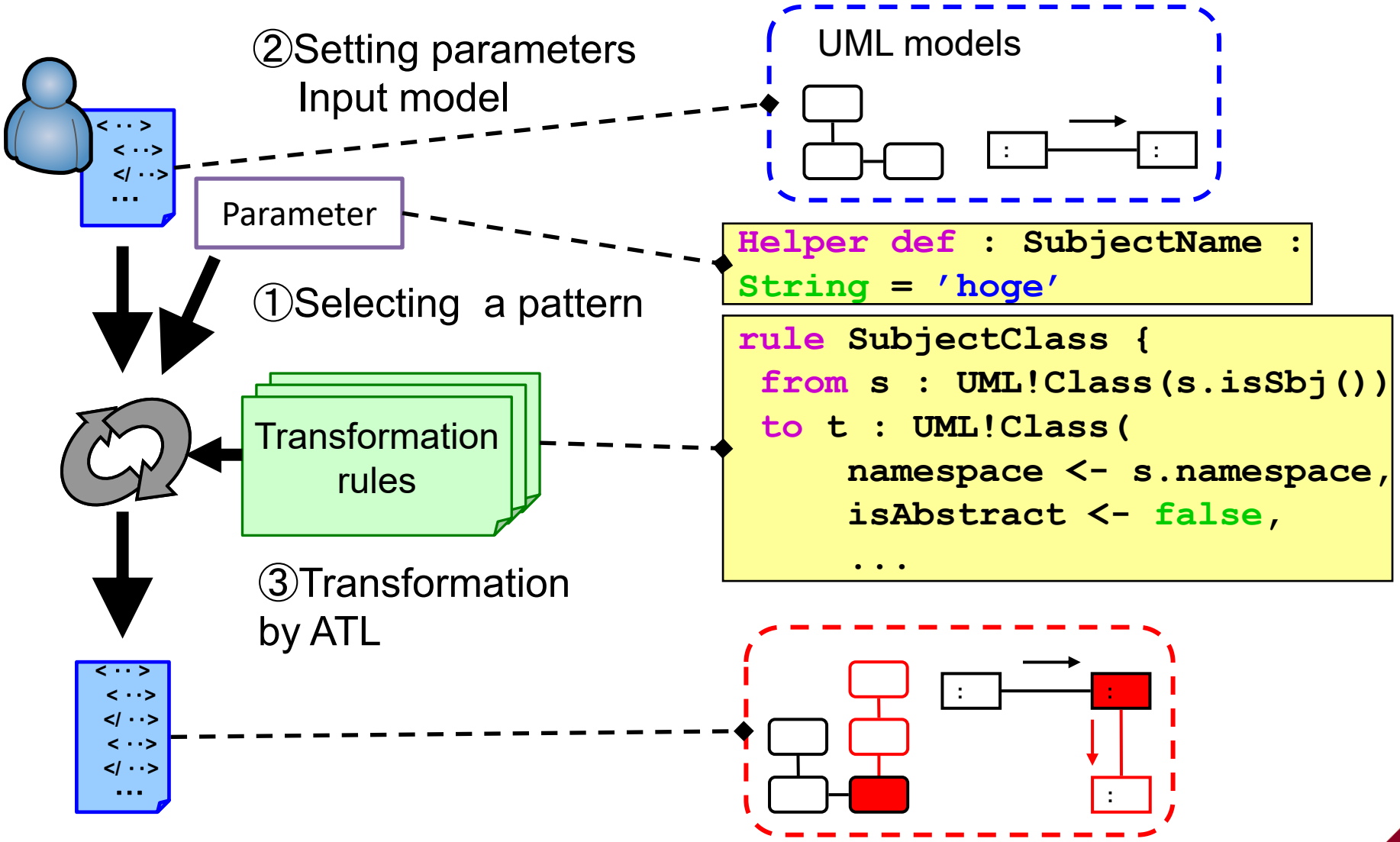
Hironori Washizaki, Tian Xia, Natsumi Kamata, Yoshiaki Fukazawa, Hideyuki Kanuka, Takehisa Kato, Masayuki Yoshino, Takao Okubo, Shinpei Ogata, Haruhiko Kaiya, Atsuo Hazeyama, Takafumi Tanaka, Nobukazu Yoshioka, G Priyalakshmi, "Systematic Literature Review of Security Pattern Research," Information, Vol. 12, No. 1:36, MDPI, pp.1-27, 2021.

Conclusion and future work

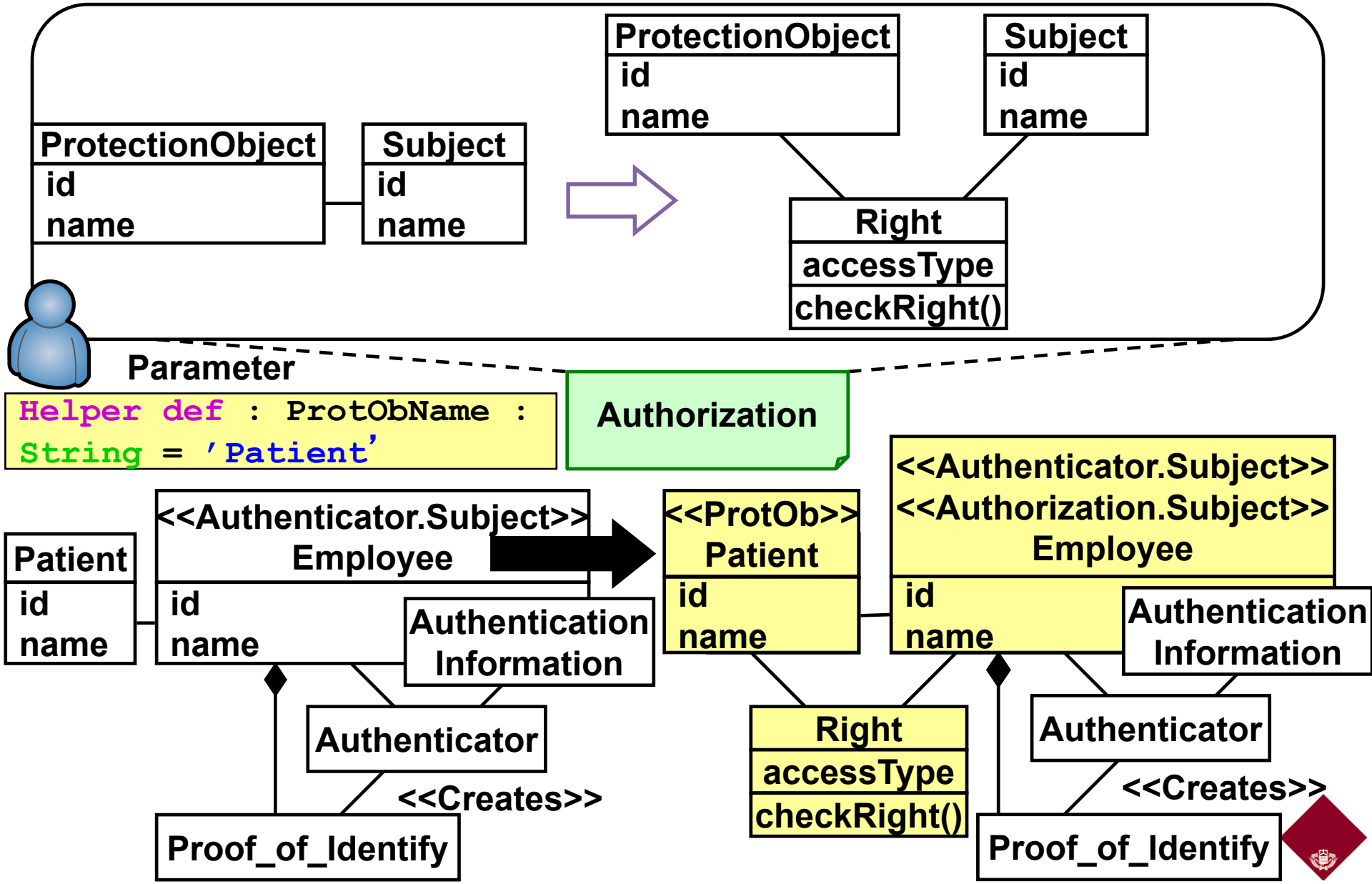
Current	Future
• Targeting authentication and authorization	• Address various security patterns
• Many researches using UML, but independent	• Integration based on common metamodel
• Often simple case studies	• Complex case studies with measurements
• Targeting existing patterns only	• New vulnerabilities and patterns
• Limited education for secure development methods in IoT era	• Cloud, IoT and security education program

Model-driven security pattern application

[PLoP'10]

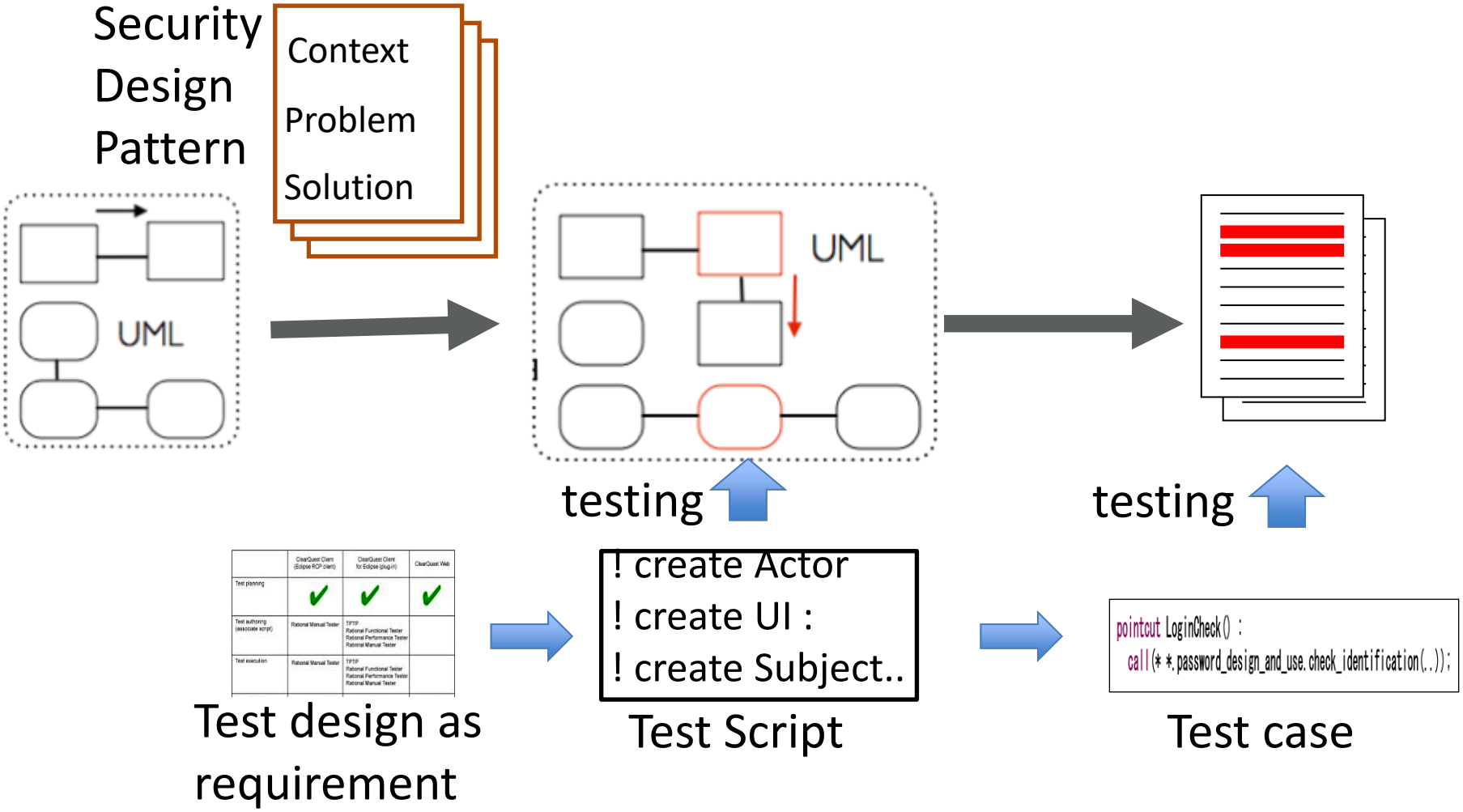


Example: application of “Authorization”



TESEM: Test Driven Secure Modeling Tool

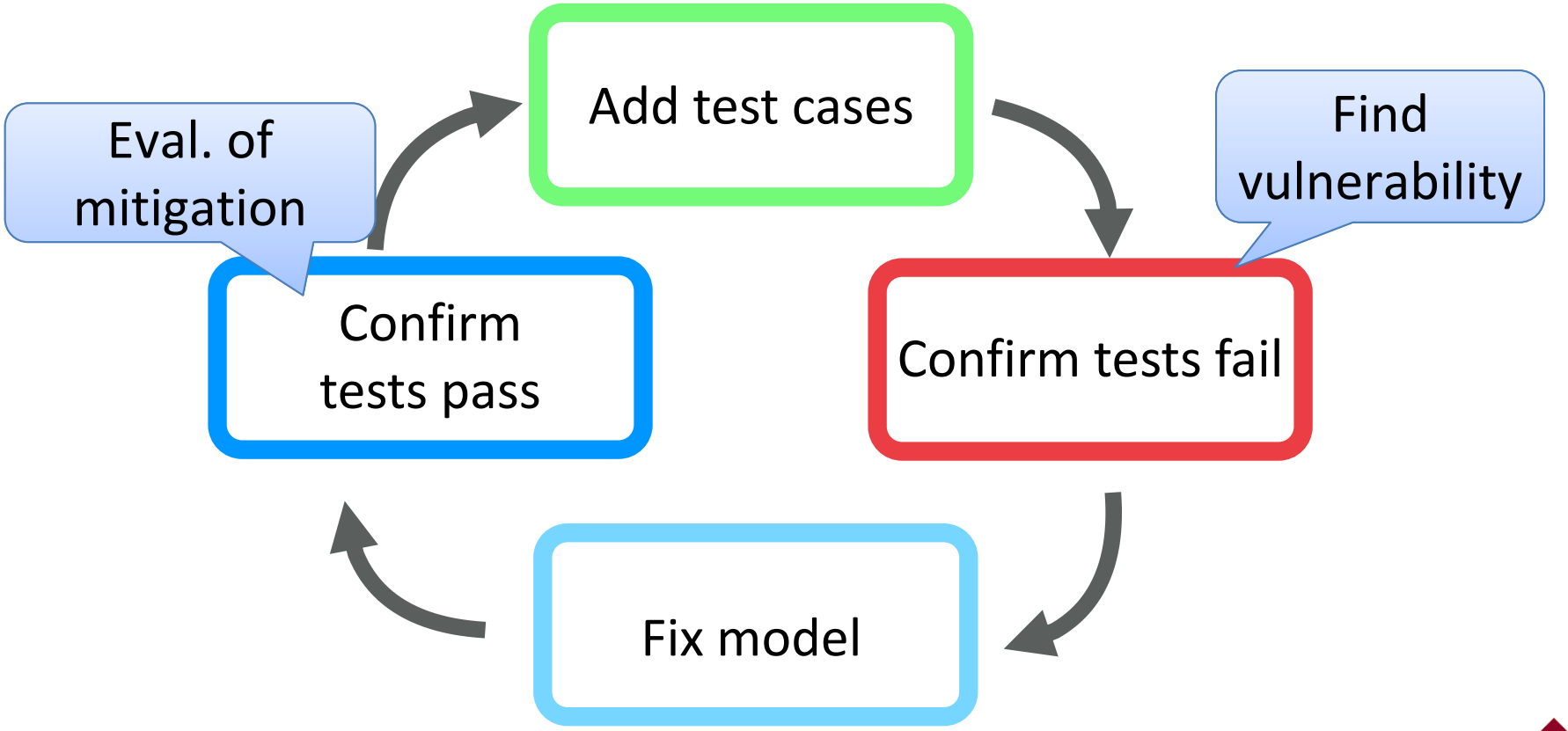
[ARES'13][ARES'13][IJSSE'14][ICST'15][Information'16]



[ARES'13] Validating Security Design Pattern Applications Using Model Testing, Int'l Conf. Availability, Reliability and Security
[ARES'14] Verification of Implementing Security Design Patterns Using a Test Template, Conf. Availability, Reliability and Security
[IJSSE'14] Validating Security Design Pattern Applications by Testing Design Models, Int'l J. Secure Software Engineering 5(4)
[ICST'15] TESEM: A Tool for Verifying Security Design Pattern Applications by Model Testing, IEEE ICST'15 Tools Track
[Information'16] Implementation Support of Security Design Patterns Using Test Templates, Information 7(2)

Test-driven secure design

- Security Properties are in testcases



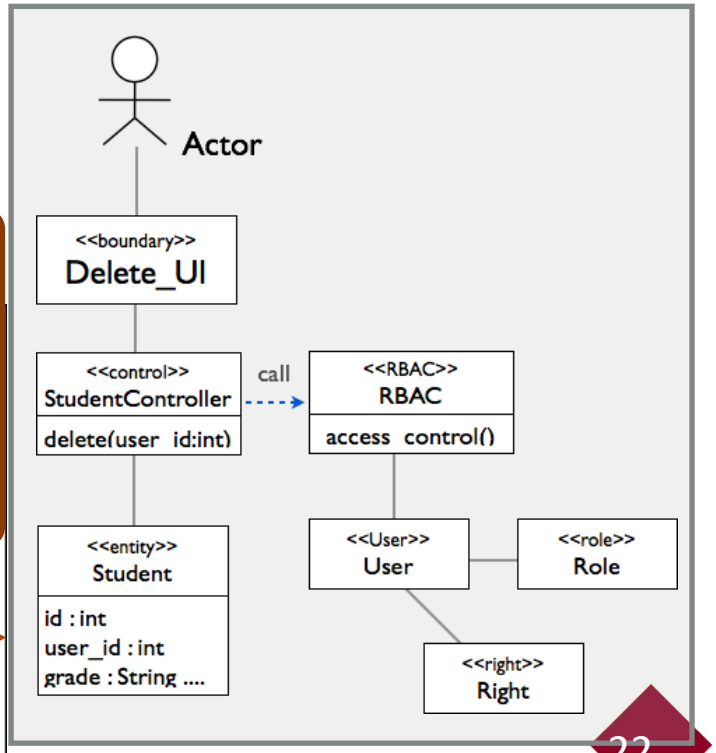
Add test cases

Verify whether model with RBAC satisfies security design requirements

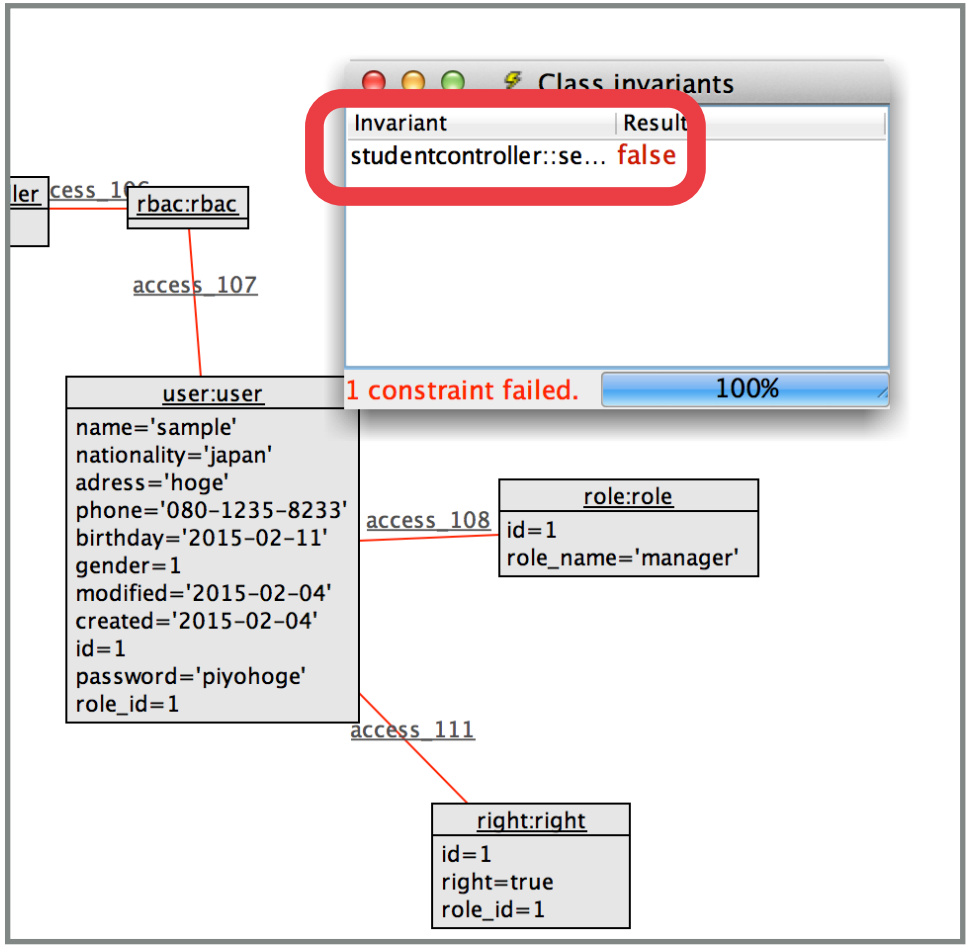
		1	2
Conditions	Rights are given in “Role” which an “User” belongs	Yes	No
Actions	consider that “Actor” have access permission.	×	
	consider that “Actor” does not have access permission.		×
	execute “delete” function	×	
	can not execute “delete” function		×

```
context subject_controller
 inv access_control:
  if self.RBAC.Right->exists(p |
        p.right = true and
        p.role_id = p.Role.id and
        p.role_id = p.Role.User.role_id )
 then
    self.DeleteUI.Actor.right = true and self.subject_function = true
 else
    self.DeleteUI.Actor.right = false and self.subject_function = false
 endif
```

Verify whether model satisfies security design requirement



Confirm tests fail



Model does not satisfy security design requirements.

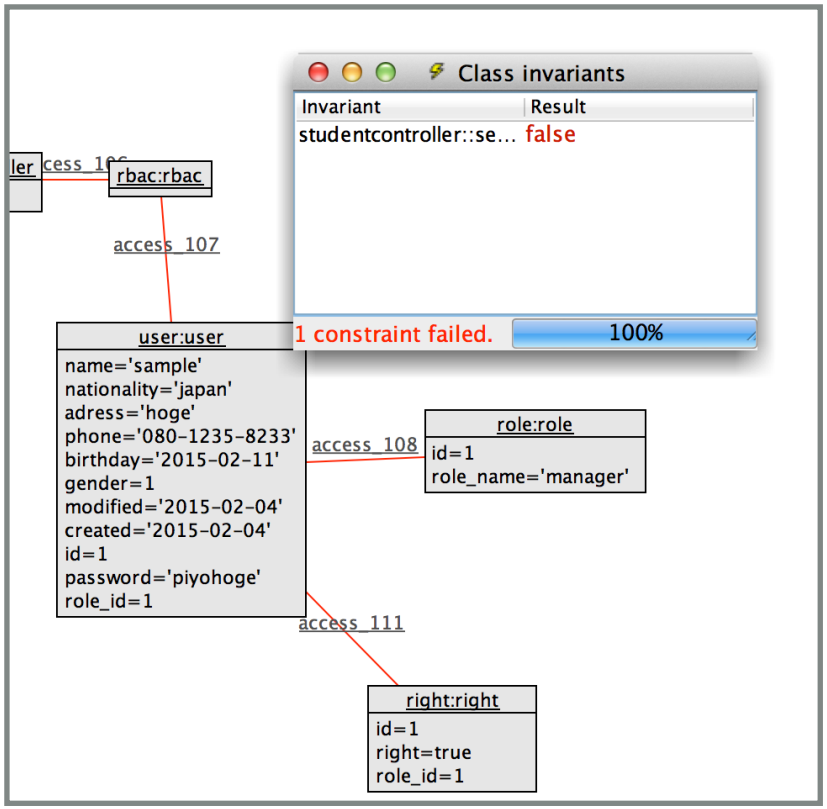
TESEM detected incorrect applications of design patterns

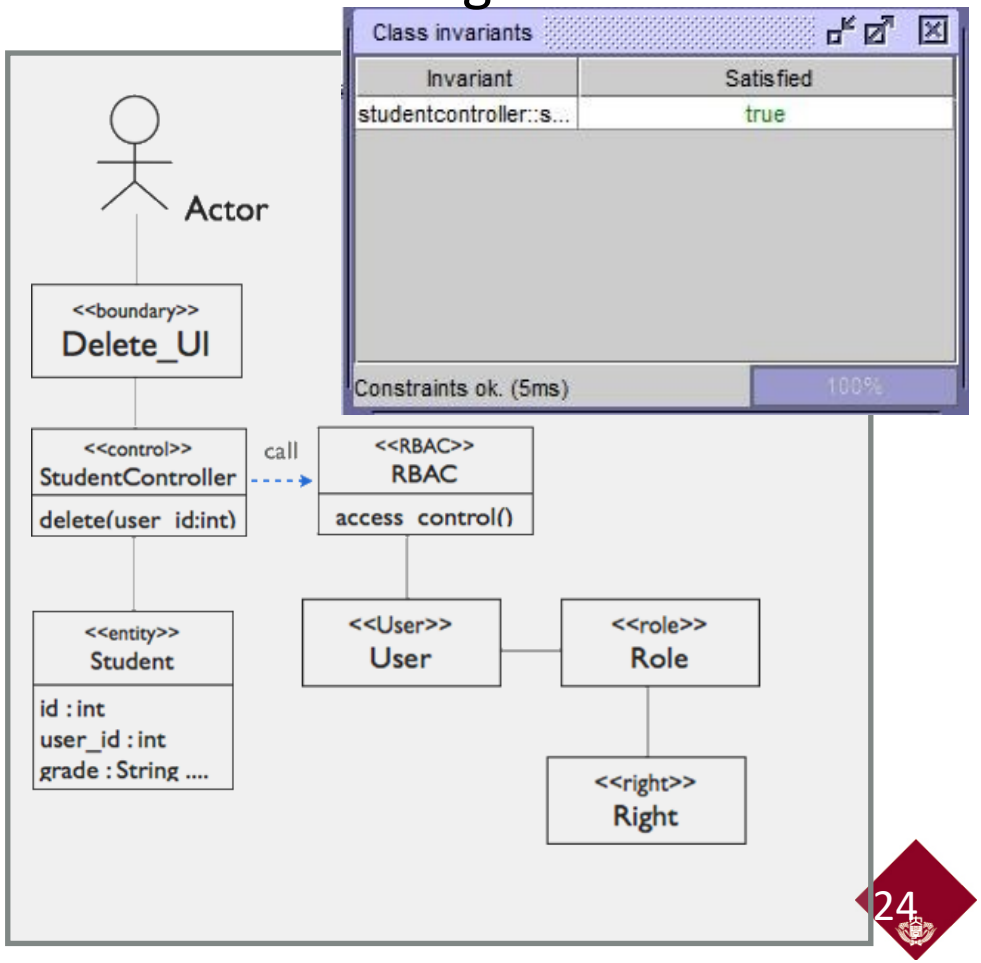
Fix model and confirm tests pass

Fix design model until the tests successfully pass.

Refactoring

Incorrect design → Correct design





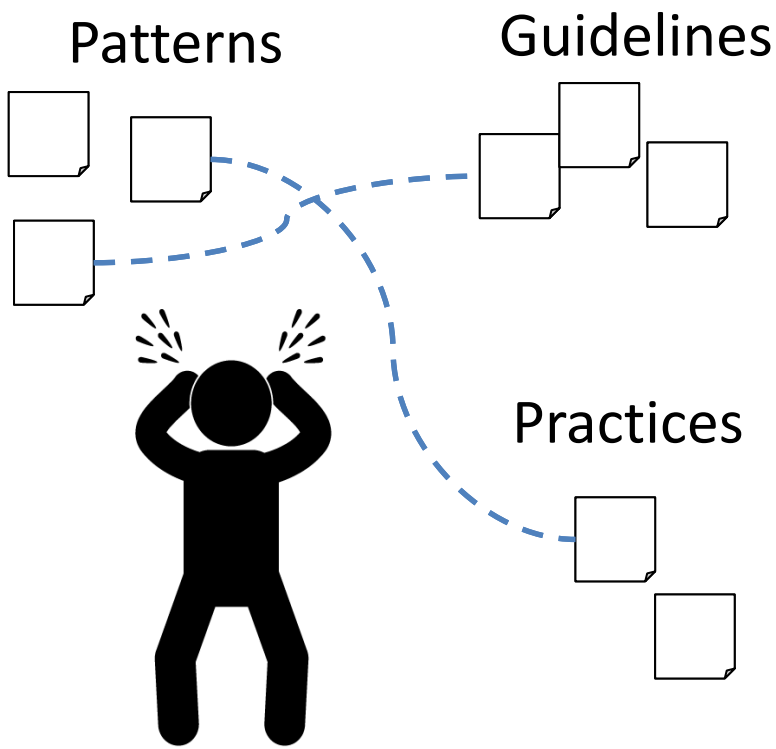
Agenda

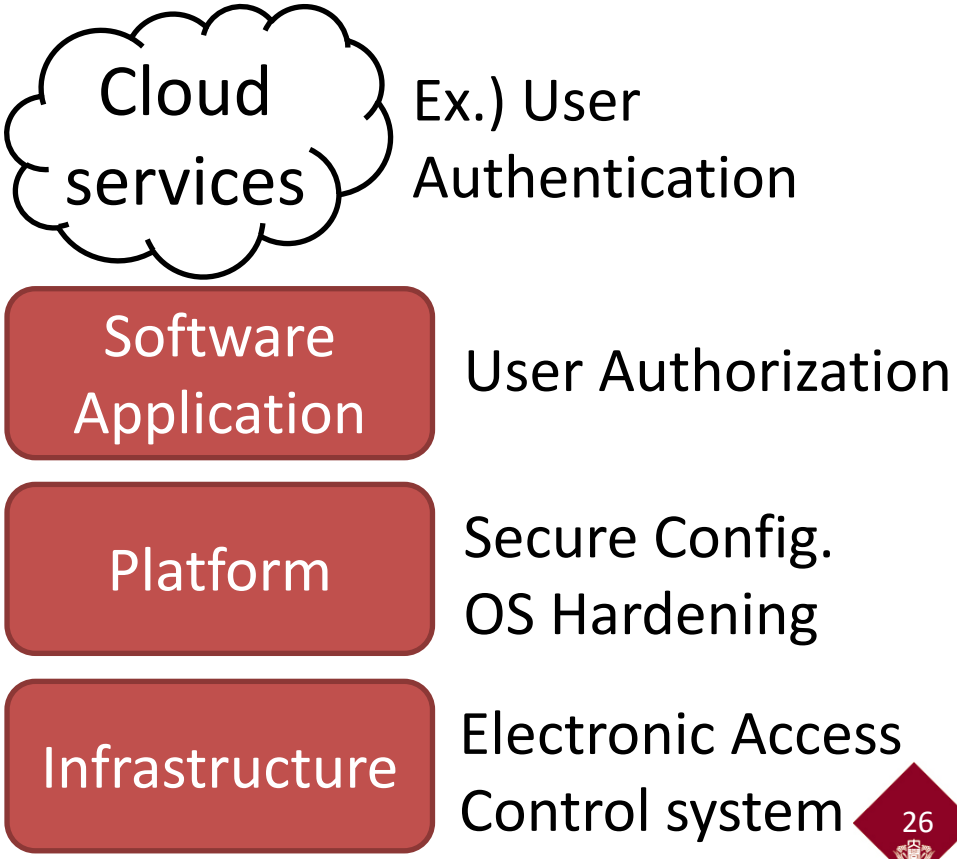
- Paradigm shifts in new software engineering
- Pattern language
- Security patterns
- Metamodel and Patterns for Cloud Security and Privacy

Challenges in cloud security and privacy (S&P)

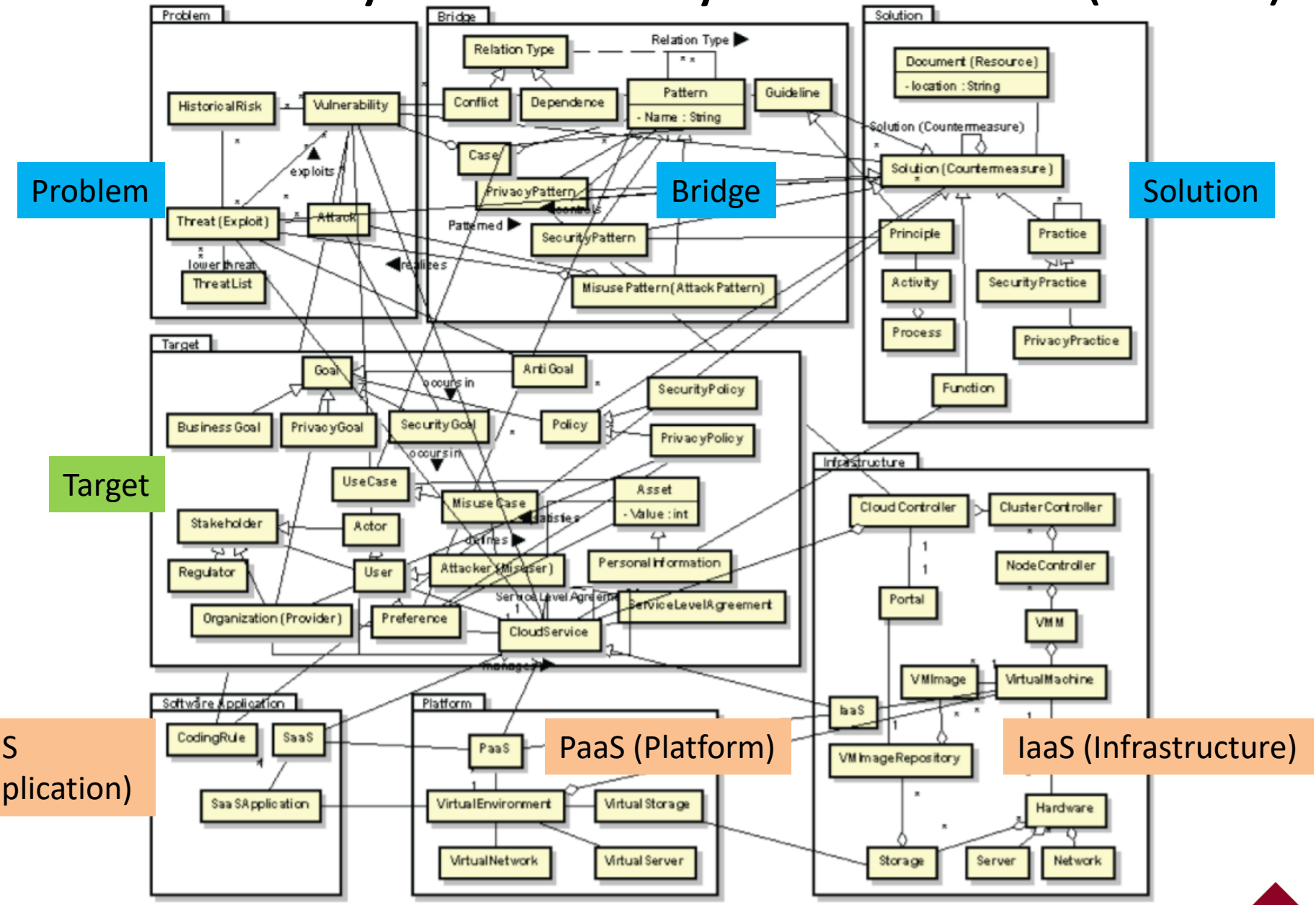
- How to consistently utilize diverse S&P knowledge?
  ⇒ Metamodel

- How to consider S&P over different layers?
  ⇒ Layered metamodel





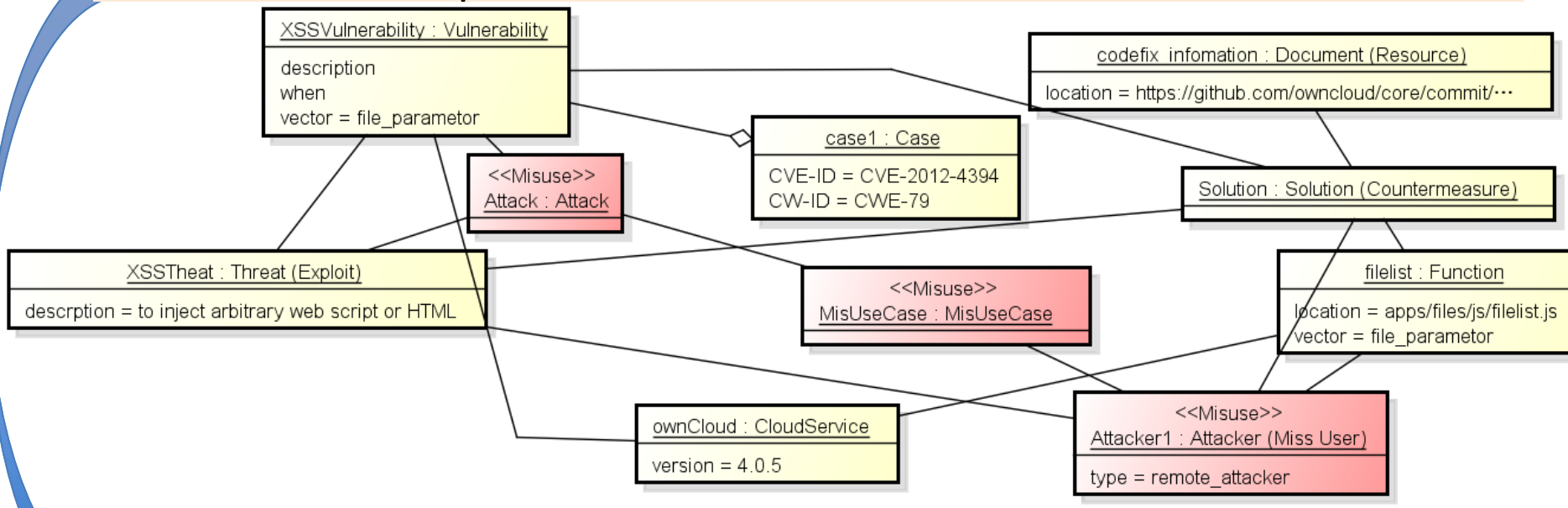
Cloud Security and Privacy Metamodel (CSPM)



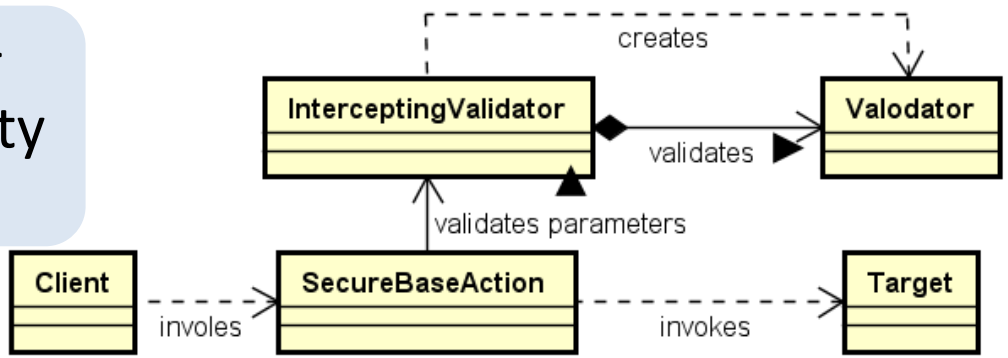
Tian Xia, Hironori Washizaki, Yoshiaki Fukazawa, Haruhiko Kaiya, Shinpei Ogata, Eduardo B. Fernandez, Takehisa Kato, Hideyuki Kanuka, Takao Okubo, Nobukazu Yoshioka and Atsuo Hazeyama, “CSPM: Metamodel for Handling Security and Privacy Knowledge in Cloud Service Development,” International Journal of Systems and Software Security and Protection (IJSSSP), Vol. 12, No. 2, IGI-Global, pp.1-18, 2021.

Modeling vulnerability and security pattern

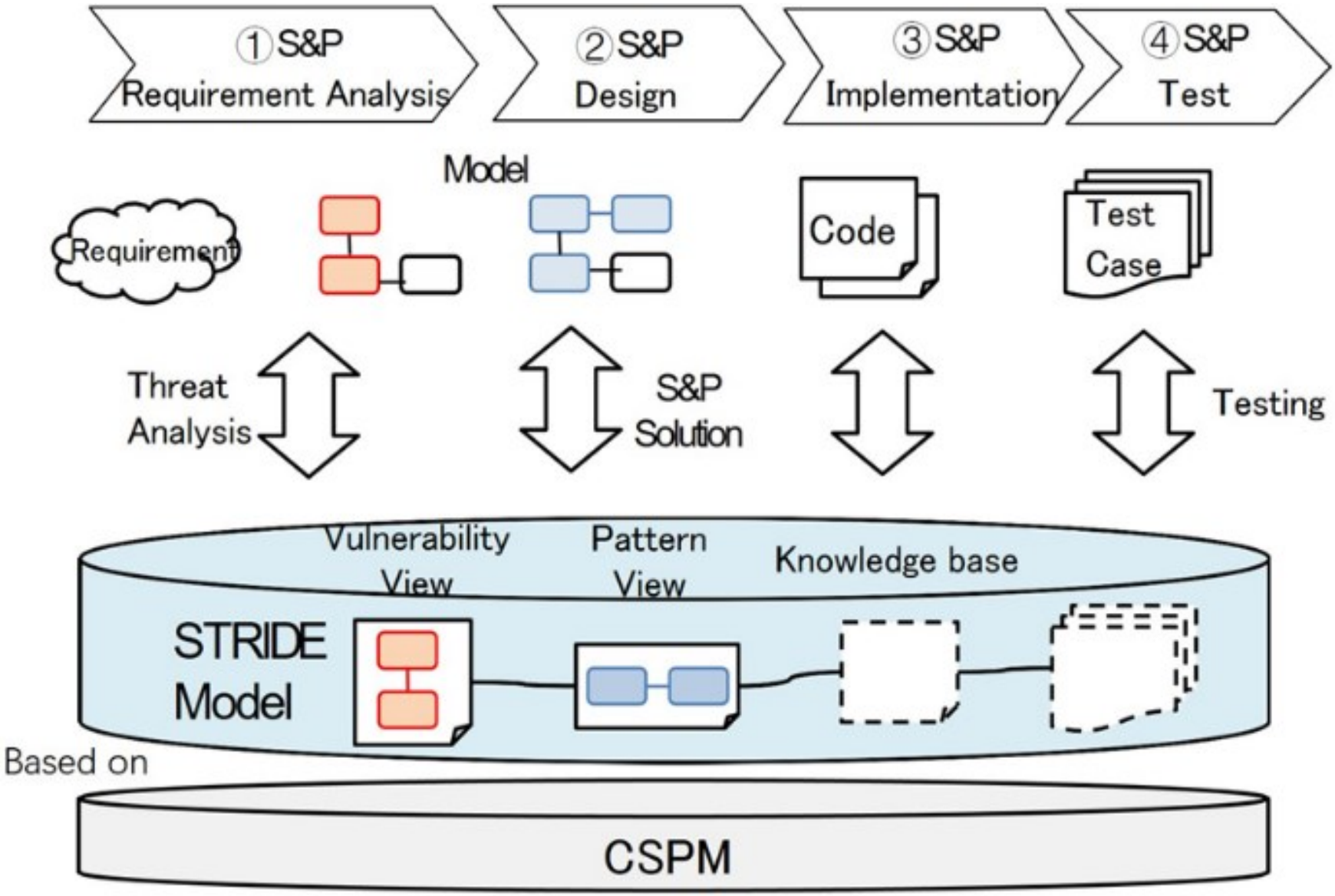
Common Vulnerabilities and Exposures: CVE-2012-4394 Cross-site scripting (XSS) vulnerability in apps/files/js/filelist.js in own Cloud before 4.0.5 allows remote attackers to inject arbitrary web script or HTML via the file parameter.



Validator for data-injection vulnerability such as XSS

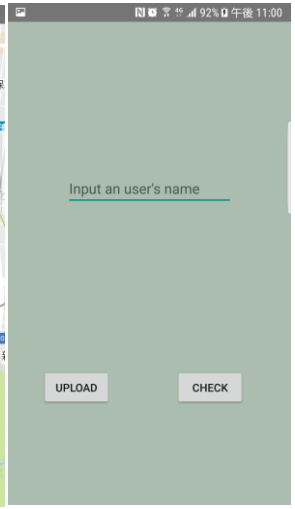


Security and privacy development process



Security requirements analysis

- Threats and vulnerability analysis based on STRIDE
- Consider corresponding security patterns (e.g., Authentication and Authorization)



Goal Anti-goal Problem Example Pattern Solution

Goal	Anti-Goal	Security Problem	Specific example	Security pattern (from implementation side)	Solution
Tamper proof data	Gain ability to tamper with data	unauthorized actors tampering with local data	users accessing local data on their phone, changing their score	Encryption pattern	Provided by the android phone itself --> it encrypts stored data
		unauthorized actors tampering with cloud data	hostile accessing the cloud server to change the goal location to current location	Encryption pattern	Handled by amazon: their security measures are quite extensive
confidentiality	Gain access to confidential information	unauthorized actors listening to the transmissions to and from the server	man in the middle attack	Transmission pattern	API automatically uses SSL and can be set to use a VPN
		information disclosure	hostile user releases a list of goal locations	Encryption pattern, Authentication and (architectural solutions: firewall, server layout	similar to tamper proof data --> same solution
		elevation of privilege	a user pretends to be an administrator which gives him unlimited access to all game data	Authentication pattern, (limitaion of access), transmission pattern	Player can only get access to the database through software, which is
non-repudiation	Gain ability to work anonymously	identity spoofing	user changes their identity and has several games running at once	Authentication pattern	Handled by API: allows users to log in using their google account
		repudiation	user able to change data anonymously making it impossible to trace	Authentication pattern	similar to identity spoofing --> same solution
Availiability	Bring down the servers	denial of service	server gets flooded by non legitimate messages meaning packets by legitimate users get dropped	firewall, patterns for Ddos	unlikely to be an issue: this game is very small scale. However, the usage of Amazon servers means that some measure of protection is in place against a DoS attack
Reliability of third-party services	Exploit usage of third party services	unsecure integration of third party services	third party authentication service is not integrated properly thus resulting in a decrease in security by making elevation of privilege easier to achieve		already handled: that is what an API is for

Spoofing

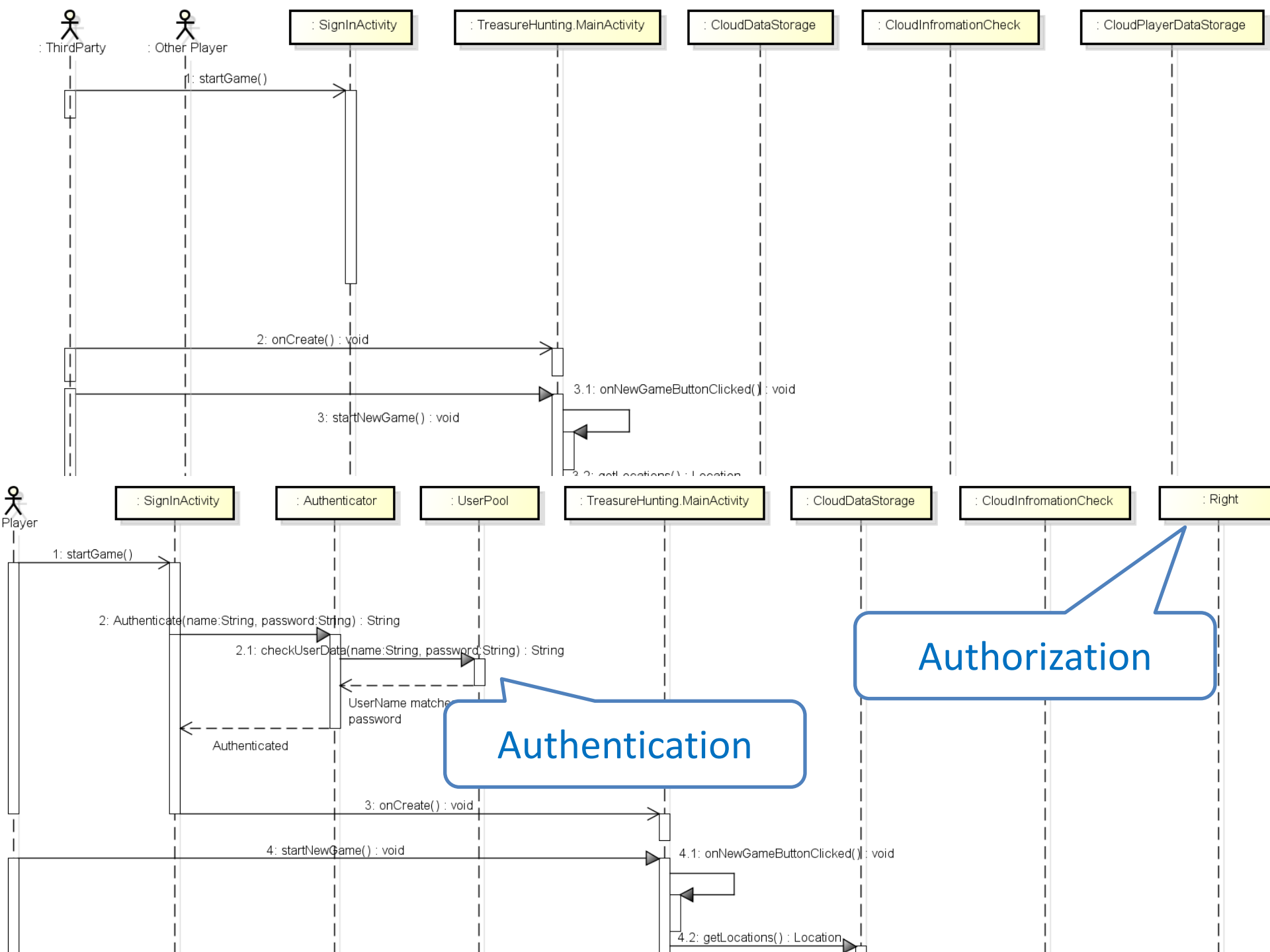
Tampering

Repudiation

Information disclosure

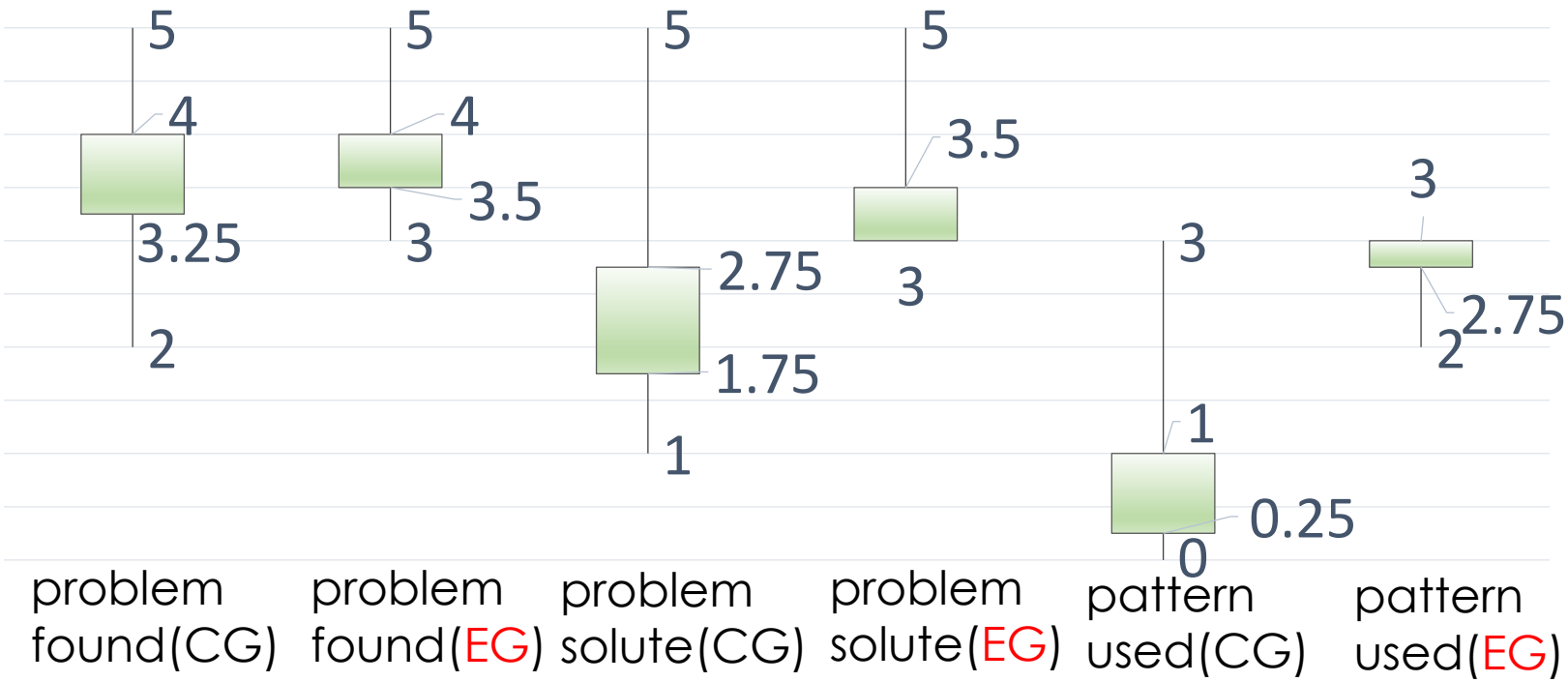
Denial of service

Elevation of privilege


: ThirdPartyPlayer
: Other Player
: SignInActivity
: TreasureHunting.MainActivity
: CloudDataStorage
: CloudInfromationCheck
: CloudPlayerDataStorage
1: startGame()
2: onCreate() : void
3.1: onNewGameButtonClicked() : void
3: startNewGame() : void
Player
: SignInActivity
: Authenticator
: UserPool
: TreasureHunting.MainActivity
: CloudDataStorage
: CloudInfromationCheck
: Right
1: startGame()
2: Authenticate(name:String, password:String) : String
2.1: checkUserData(name:String, password:String) : String
UserName matches
password
Authenticated
Authentication
Authorization
3: onCreate() : void
4: startNewGame() : void
4.1: onNewGameButtonClicked() : void
4.2: getLocations() : Location

Case study and result

- This table shows the result of a simple case study by assigning a vulnerable system model.
- The experiment group with CSPM perform better by solving more security problems.



Summary

- There are paradigm shifts in “new” software engineering.
  - Cloud computing is one of the key enablers of digital transformations.
  - Security must be a critical cross-cutting concern in cloud and any other software.
- New software engineering needs patterns and pattern languages.
  - Bridge between abstract paradigms and concrete cases/tools
  - Common language among stakeholders
- Security patterns
  - Systematic Literature Review of Security Pattern Research
  - Model-driven security pattern application
  - Test Driven Secure Modeling Tool
- Metamodel and Patterns for Cloud Security and Privacy
  - Cloud Security and Privacy Metamodel (CSPM)
  - Modeling vulnerability and security pattern
  - Security and privacy development process