The Seventh International Conference on Fundamentals and Advances in Software Systems Integration (FASSI 2021)
https://www.iaria.org/conferences2021/FASSI21.html
November 14-18, 2021 Athens, Greece

Patterns for New Software Engineering: Machine Learning, IoT and Security Patterns

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
  - **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

Patterns for New Software Engineering: Machine Learning, IoT and Security Patterns

Hironori Washizaki

Professor at Waseda University, Tokyo, Japan

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

- Jomphon Runpakprakun, Sien Reeve Ordonez Peralta, Hironori Washizaki, Foutse Khomh, Yann-Gael Gueheneuc, Nobukazu Yoshioka, Yoshiaki Fukazawa, "Software Engineering Patterns for Machine Learning Applications (SEP4MLA) – Part 3 – Data Processing Architectures," 28th Conference on Pattern Languages of Programs (PLoP 2021), pp. 1-11, 2021.
- Hironori Washizaki, Hironori Takeuchi, Foutse Khomh, Naotake Natori, Takuo Doi, Satoshi Okuda, "Practitioners' insights on machine-learning software engineering design patterns: a preliminary study," 36th IEEE International Conference on Software Maintenance and Evolution (ICSME 2020), Late Breaking Ideas track
- Hironori Washizaki, Shinpei Ogata, Atsuo Hazeyama, Takao Okubo, Eduardo B. Fernandez, Nobukazu Yoshioka, "Landscape of Architecture and Design Patterns for IoT Systems," IEEE Internet of Things Journal, Vol. 7, No. 10, pp.10091 – 10101, 2020
- Eduardo B. Fernandez, Hironori Washizaki, Nobukazu Yoshioka, Takao Okubo, "The design of secure IoT applications using patterns: State of the art and directions for research," Internet of Things; Engineering Cyber Physical Human Systems, Vol. 15-16, Elsevier, pp. 1-25, 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.
- 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.

Supported by JST MIRAI eAI JPMJMI20B8 https://www.jst.go.jp/mirai/jp/program/super-smart/JPMJMI20B8.html
Supported by JSPS Bilateral Joint Research Project

Agenda

- Paradigm shifts in new software engineering
- Pattern language
- Machine learning patterns
- IoT patterns
- Security patterns

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

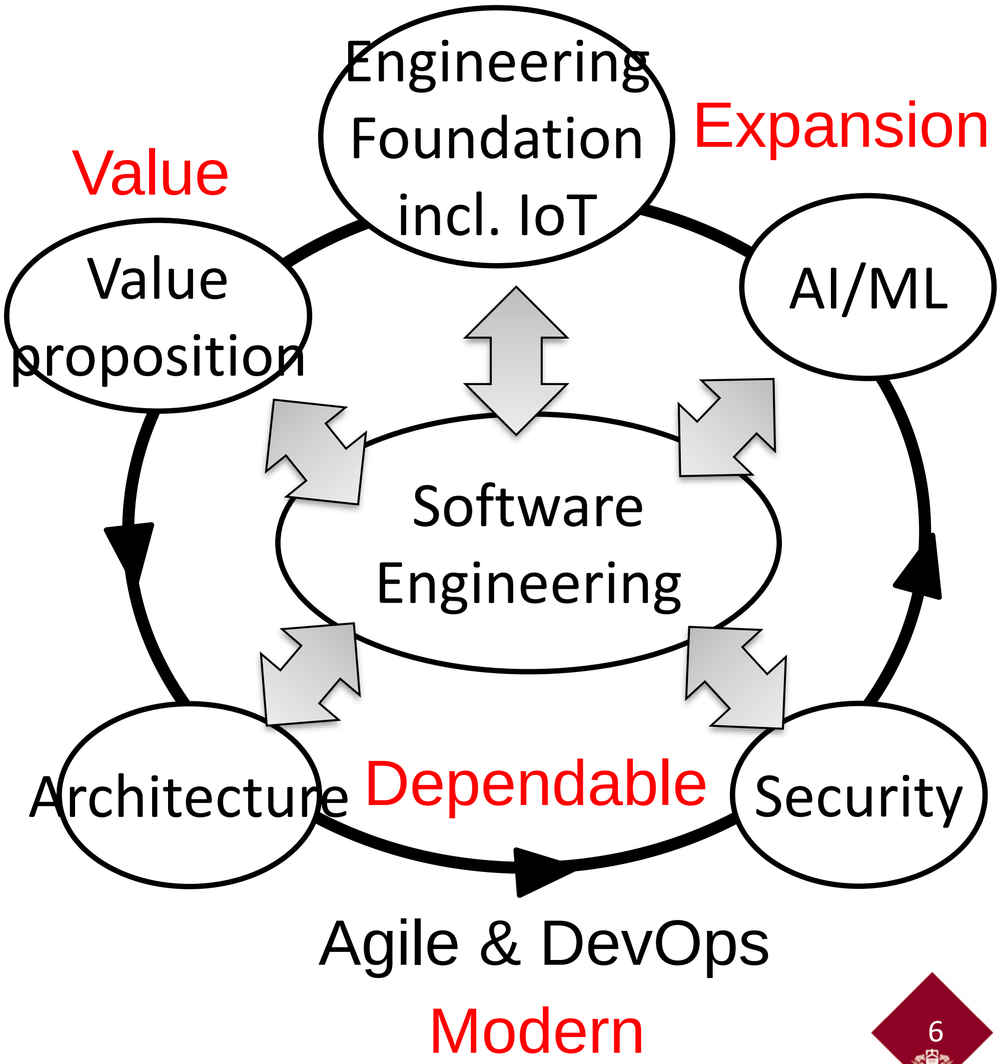
- Computing Foundations
- Mathematical Foundations
- Engineering Foundations

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



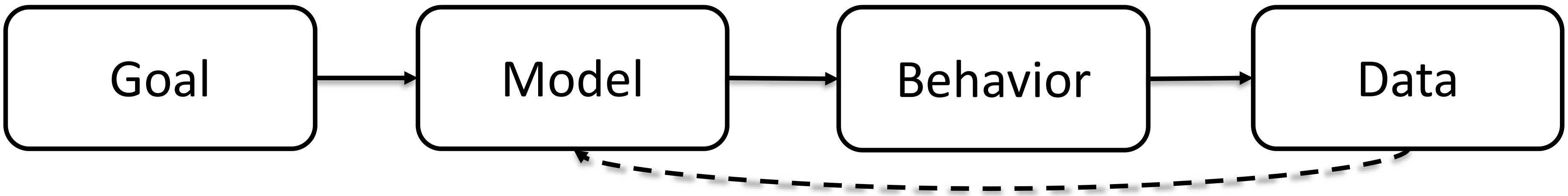
Paradigm shifts in “new” software engineering

	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**

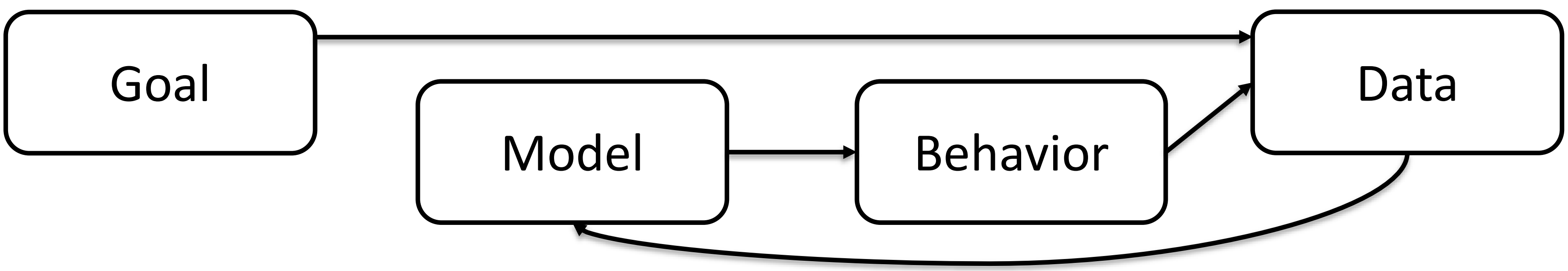
Hironori Washizaki, Junzo Hagimoto, Kazuo Hamai, Mitsunori Seki, Takeshi Inoue, Shinya Taniguchi, Hiroshi Kobayashi, Kenji Hiranabe and Eiichi Hanyuda, “Framework and Value-Driven Process of Software Engineering for Business and Society (SE4BS),” 5th International Conference on Enterprise Architecture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

Example: Deduction and induction

Conventional software: Deduction



ML-based software: Induction

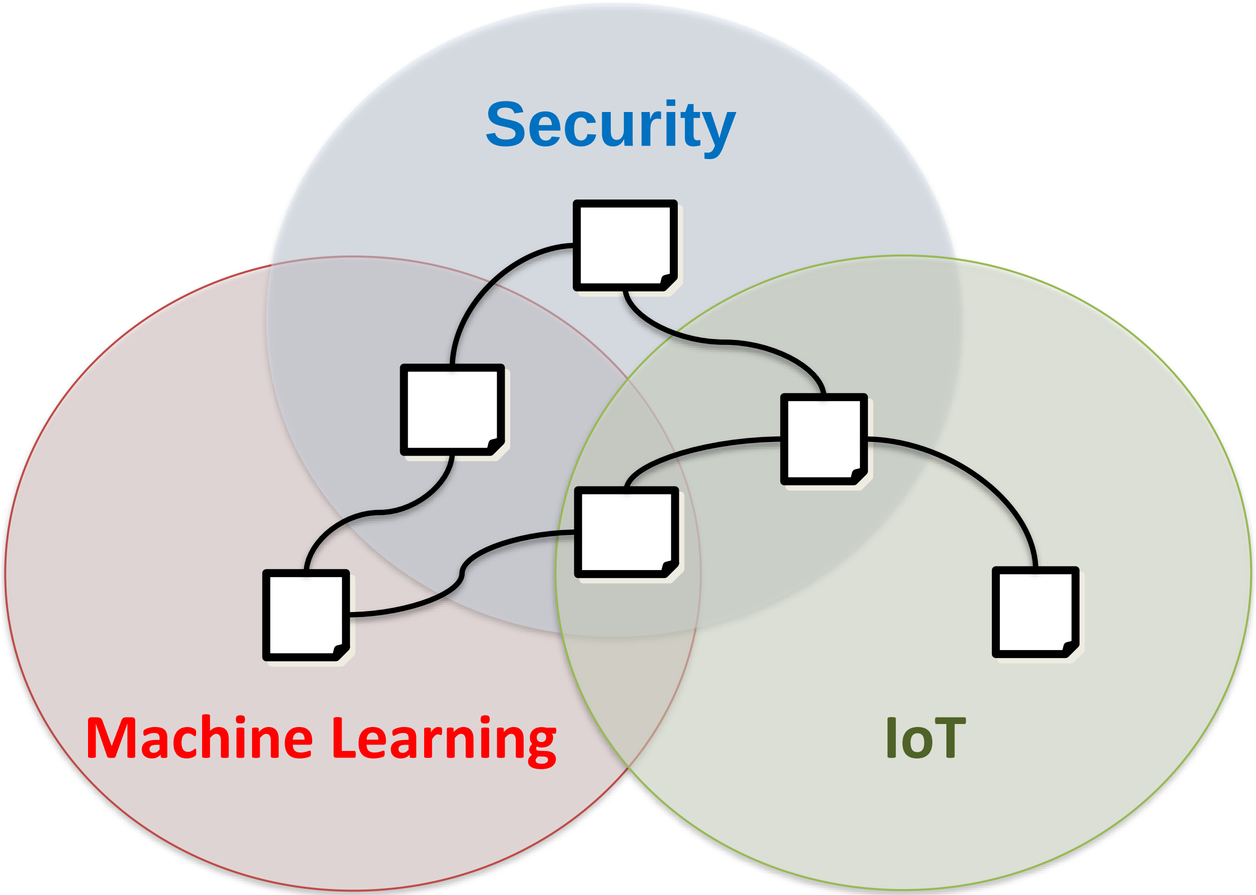


H. Maruyama, “Machine Learning Engineering and Reuse of AI Work Products,” The First International Workshop on Sharing and Reuse of AI Work Products, 2017

Hironori Washizaki, “Towards Software Value Co-Creation with AI”, The 44th IEEE Computer Society Signature Conference on Computers, Software, and Applications (COMPSAC 2020), Fast Abstract

Problem and goal

- ML and IoT are key enablers of digital transformations.
- Patterns in ML and IoT software design are not well classified and studied.
- Security must be a critical cross-cutting concern in ML and IoT software.
- We are conducting a systematic literature review to reveal landscapes of ML, IoT, and security software engineering patterns.



Agenda

- Paradigm shifts in new software engineering
- Pattern language
- Machine learning patterns
- IoT patterns
- Security patterns

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/8IKf54pc3qk

https://unsplash.com/photos/zACLEreWKXE

Towards a pattern language

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

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

Small Public Square

https://unsplash.com/photos/zFoRwZirFvY

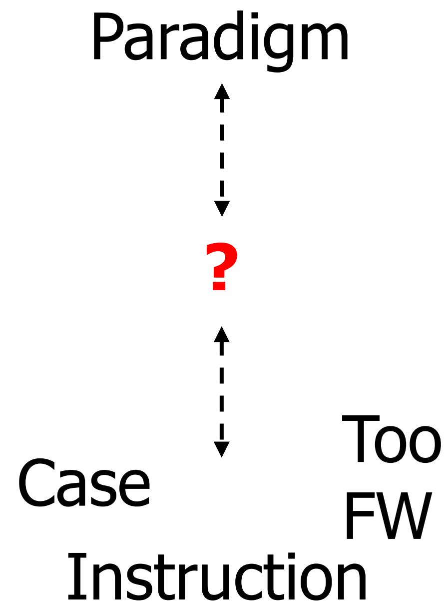
Street Cafe

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
  – Software engineers, hardware engineers, network engineers, domain experts, data analysist …

Agenda

- Paradigm shifts in new software engineering
- Pattern language
- Machine learning patterns
- IoT patterns
- Security patterns

Practices and patterns in ML-SE

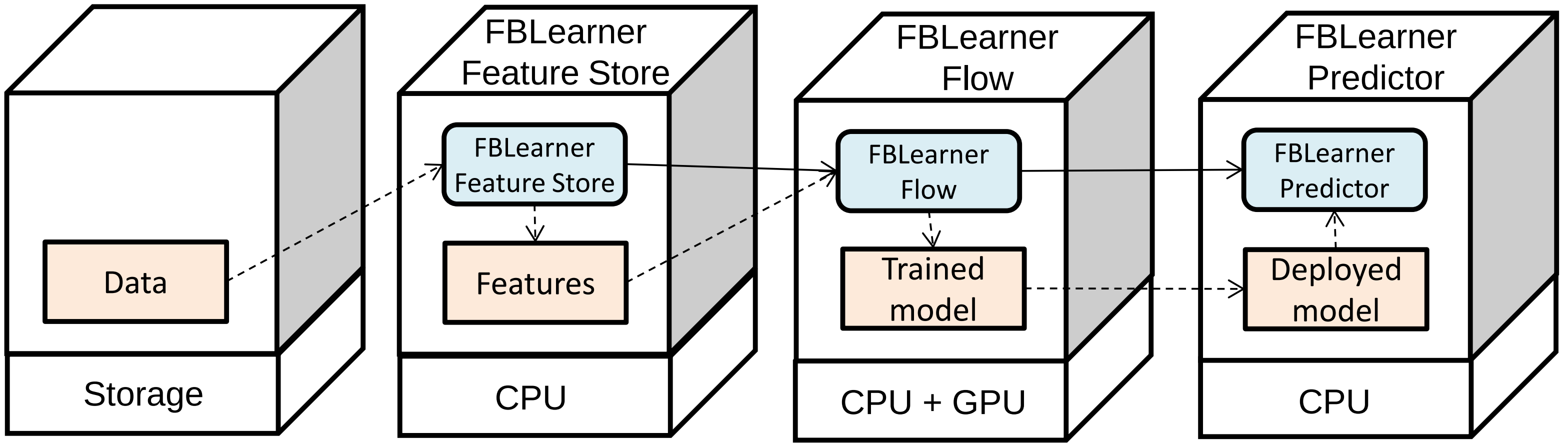
Data Lake for ML

https://unsplash.com/photos/_HPlEmsKgP0

- Researchers and practitioners studying best practices strive to design Machine Learning (ML) systems and software.
- Some practices are formalized as patterns.

(NOTE: NOT handle ML model patterns.)

Different Workloads in Different Computing Environments (e.g., Facebook)



K. M. Hazelwood, et al., Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective, HPCA 2018
H. Washizaki, et al. Software Engineering Patterns for Machine Learning Applications (SEP4MLA) – Part 2, PLoP'20

Research questions

- **RQ1. Does academic and gray literature address the design of ML systems and software?**
  - 19 scholarly and 19 gray documents identified
  - 15 SE patterns for ML applications extracted
- **RQ2. Can ML patterns be classified?**
  - Categories of scopes: Topology, programming and model
  - Quality attributes: ISO/IEC 25010:2011 System and software product quality attributes, ML model and prediction quality attributes
- **RQ3. How do practitioners perceive ML patterns?**
  - Questionnaire-based survey for 600+ developers
  - Developers were unfamiliar with most ML patterns, although there were several major patterns used by 20%

RQ1. Does academic and gray literature address the design of ML systems and software?

- Systematic Literature Review (SLR)
  - Scholar papers: Engineering Village
  - Gray documents: Google
- 19 scholarly papers and 19 gray documents identified
- 15 patterns extracted

Engineering Village

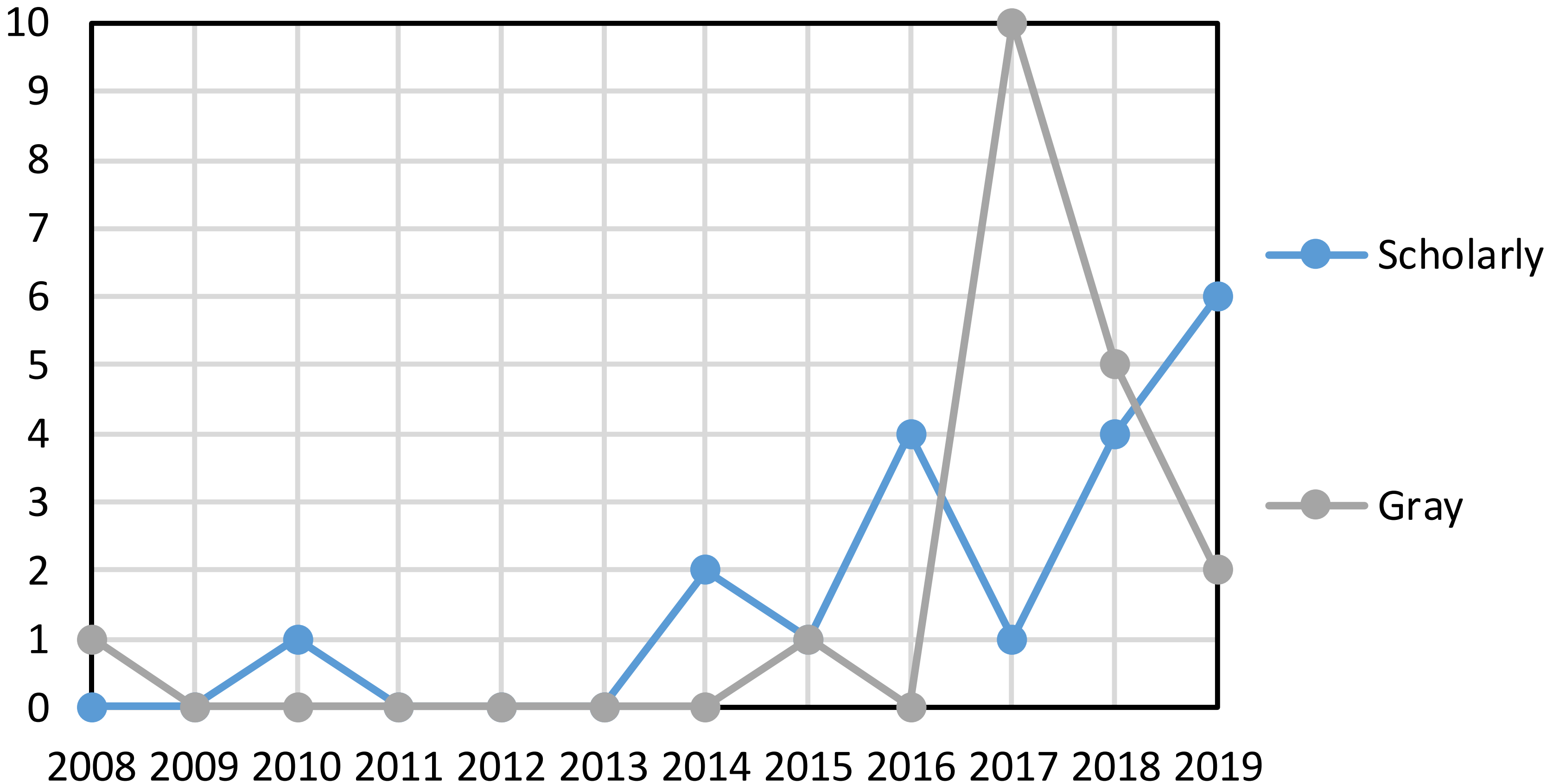
((((system) OR (software)) AND (machine learning) AND (implementation pattern) OR (pattern) OR (architecture pattern) OR (design pattern) OR (anti-pattern) OR (recipe) OR (workflow) OR (practice) OR (issue) OR (template))) WN ALL) + ((cpx OR ins OR kna) WN DB) AND (({ca} OR {ja} OR {ip} OR {ch}) WN DT)

Google

(system OR software) "Machine learning" (pattern OR "implementation pattern" OR "architecture pattern" OR "design pattern“ OR anti-pattern OR recipe OR workflow OR practice OR issue OR template)
"machine implementation pattern" OR "architecture pattern" OR "design pattern“ OR anti-pattern OR recipe OR workflow OR practice OR issue OR template

Numbers of Documents per Year

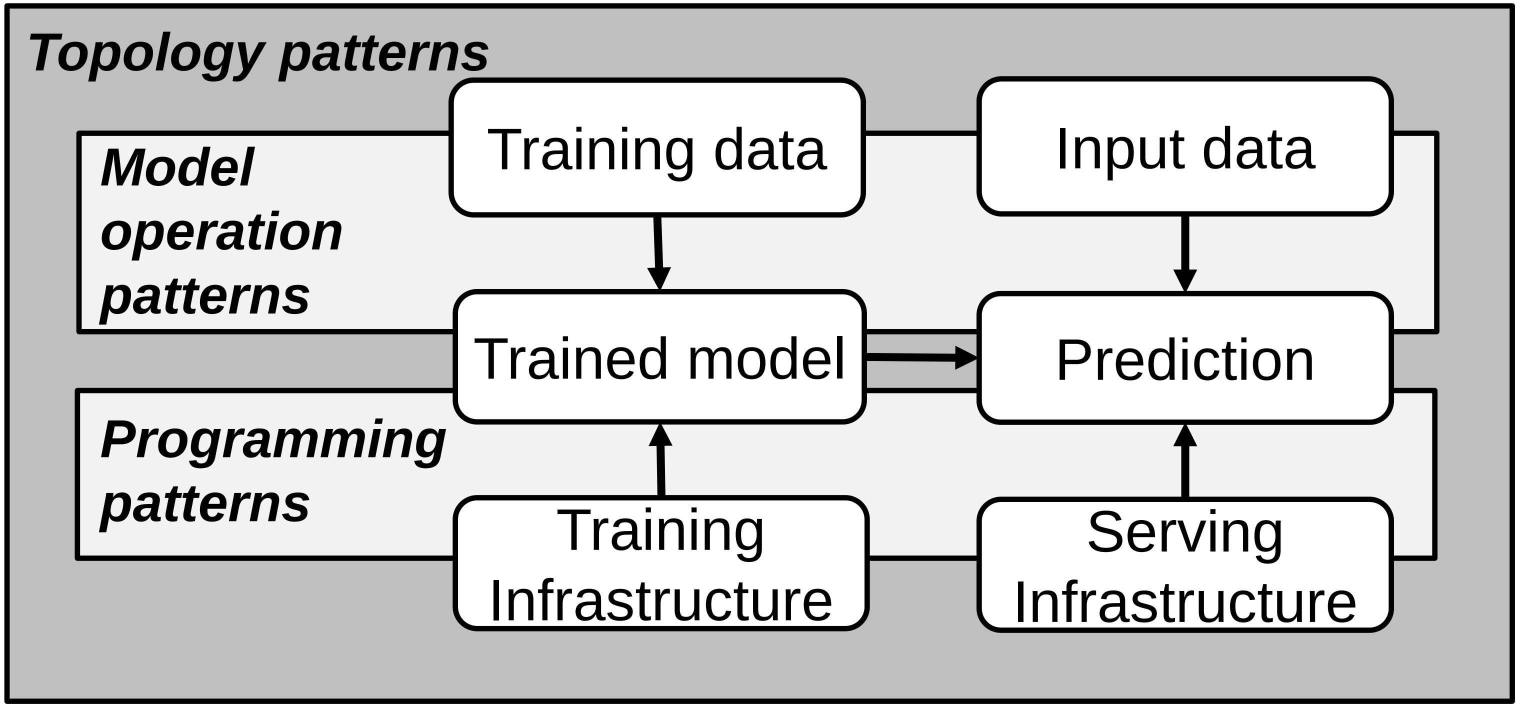
- ML application systems have recently become popular due to the promotion of artificial intelligence.
- Since 2008, academic and gray documents have discussed good (bad) practices of ML application systems design.



H. Washizaki, et al., Studying Software Engineering Patterns for Designing Machine Learning Systems, IWESEP 2019

RQ2. Can ML patterns be classified?

- Model operation patterns that focus on ML models
- Programming patterns that define the design of a particular component
- Topology patterns that define the entire system architecture.



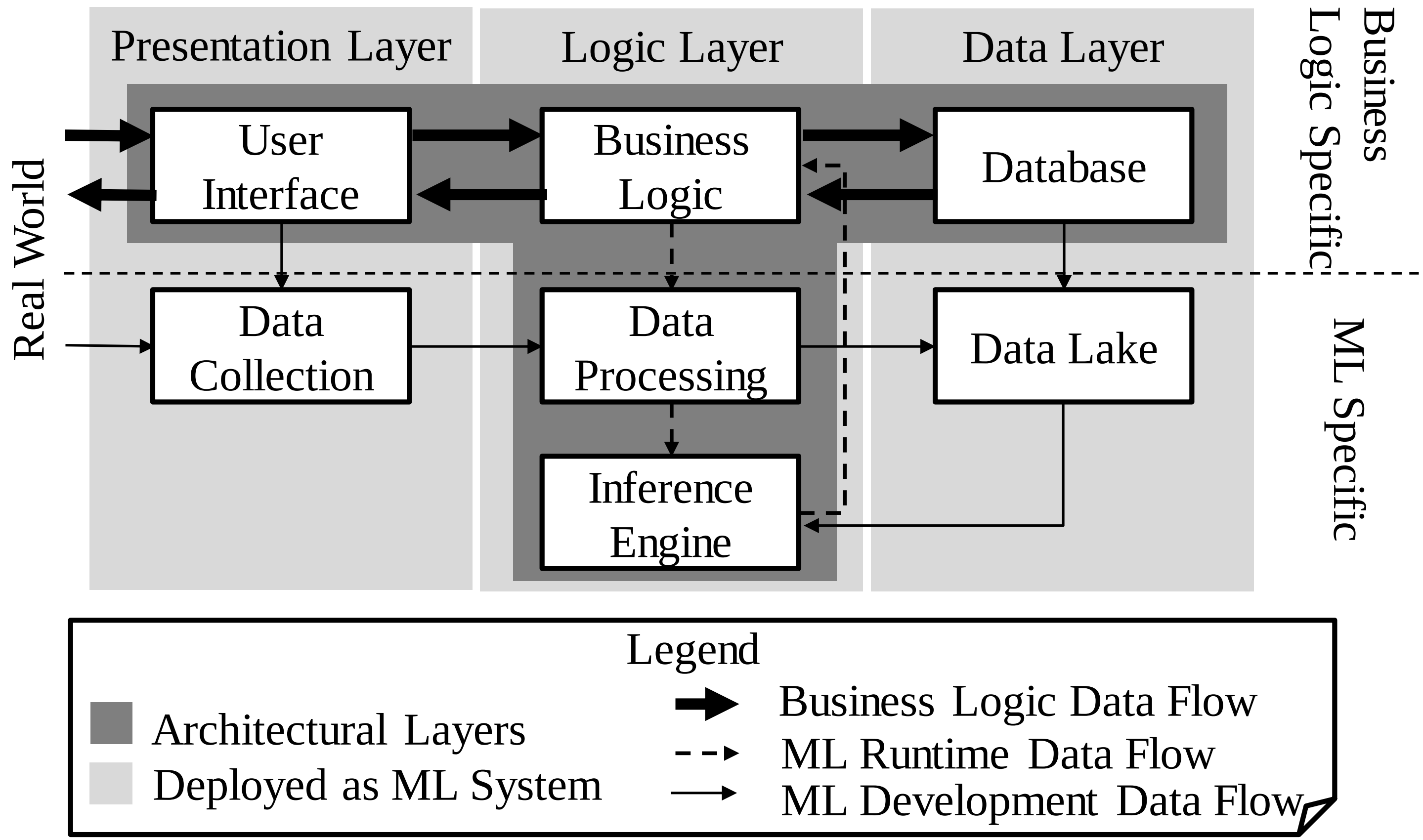
H. Washizaki, et al., Practitioners' insights on machine-learning software engineering design patterns: a preliminary study, ICSME 2020

Topology patterns

Pattern	Problem	Solution
Different Workloads in Different Computing Environments	It is necessary to separate and quickly change the ML data workload …	Physically isolate different workloads to separate machines…
Distinguish Business Logic from ML Models	The overall business logic should be isolated from the ML models …	Separate the business logic and the inference engine, loosely coupling the business logic and ML-specific dataflows.
ML Gateway Routing Architecture	Difficult to set up and manage individual endpoints for each service…	Install a gateway before a set of applications …
Microservice Architecture for ML	ML applications may be confined to some “known” ML frameworks …	Provide well-defined services to use for ML frameworks….
Lambda Architecture for ML	Real-time data processing requires scalability, fault tolerance, predictability …	The batch layer keeps producing views while the speed layer creates the relevant real-time views …
Kappa Architecture for ML	It is necessary to deal with huge amount of data with less code resource …	Support both real-time data processing and continuous reprocessing with a single stream processing engine …

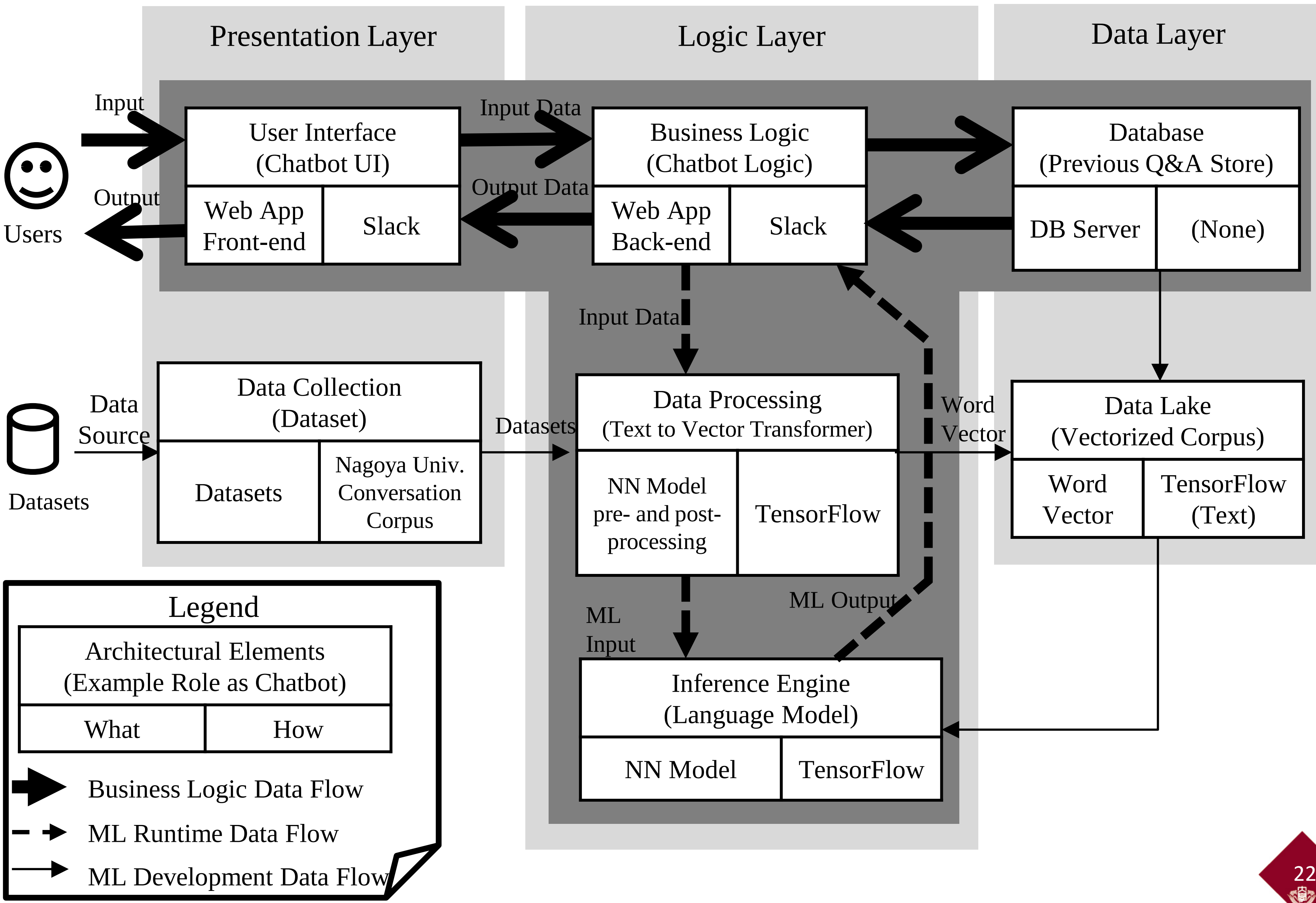
Distinguish Business Logic from ML Models

- **Problem:** Business logic should be isolated from ML models so that they can be changed without impacting rest of business logic.
- **Solution:** Separate the business logic and the inference engine, loosely coupling the business logic and ML-specific dataflows.



H. Yokoyama, Machine Learning System Architectural Pattern for Improving Operational Stability, ICSA-C, 2019
H. Washizaki, et al., Software Engineering Patterns for Machine Learning Applications (SEP4MLA), AsianPLoP 2020

Usage of **Distinguish Business Logic from ML Models**

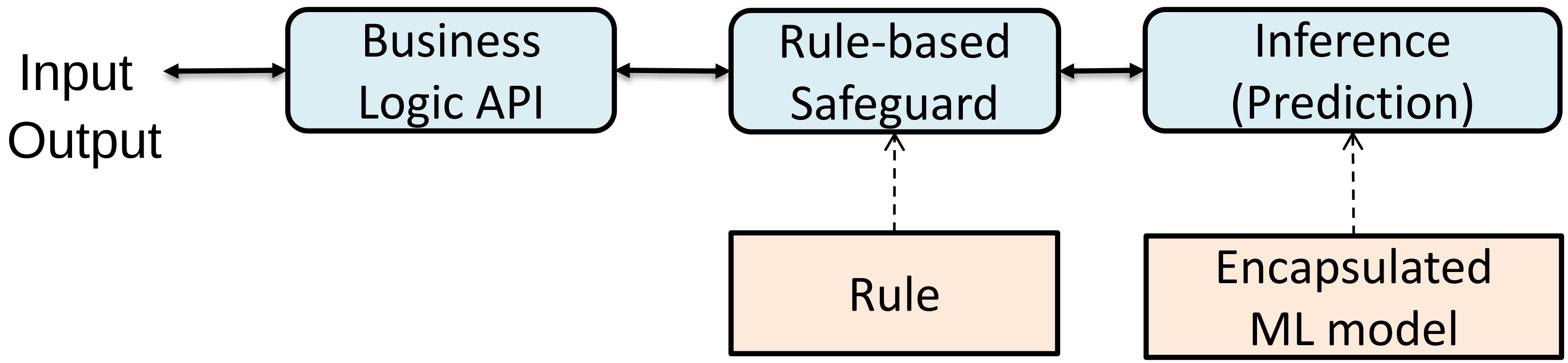


Programming patterns

Pattern	Problem	Solution
Data Lake for ML	We cannot foresee the kind of analyses that will be performed on the data …	Store data, which range from structured to unstructured, as “raw” as possible into a data storage …
Separation of Concerns and Modularization of ML Components	ML applications must accommodate regular and frequent changes to their ML components …	Decouple at different levels of complexity from the simplest to the most complex …
Encapsulate ML Models within Rule-based Safeguards	ML models are known to be unstable and vulnerable to adversarial attacks, drifts, …	Encapsulate functionality in the containing system using deterministic and verifiable rules …
Discard PoC Code	The code created for Proof of Concept (PoC) often includes code that sacrifices maintainability …	Discard the code created for the PoC and rebuild maintainable code …

Encapsulate ML Models within Rule-based Safeguards

- **Problem:** ML models are known to be unstable and vulnerable to adversarial attacks, noise, and data drift.
- **Solution:** Encapsulate functionality provided by ML models and deal with the inherent uncertainty in the containing system using deterministic and verifiable rules.
- Know usage: E.g. Apollos's object detection [Peng20]



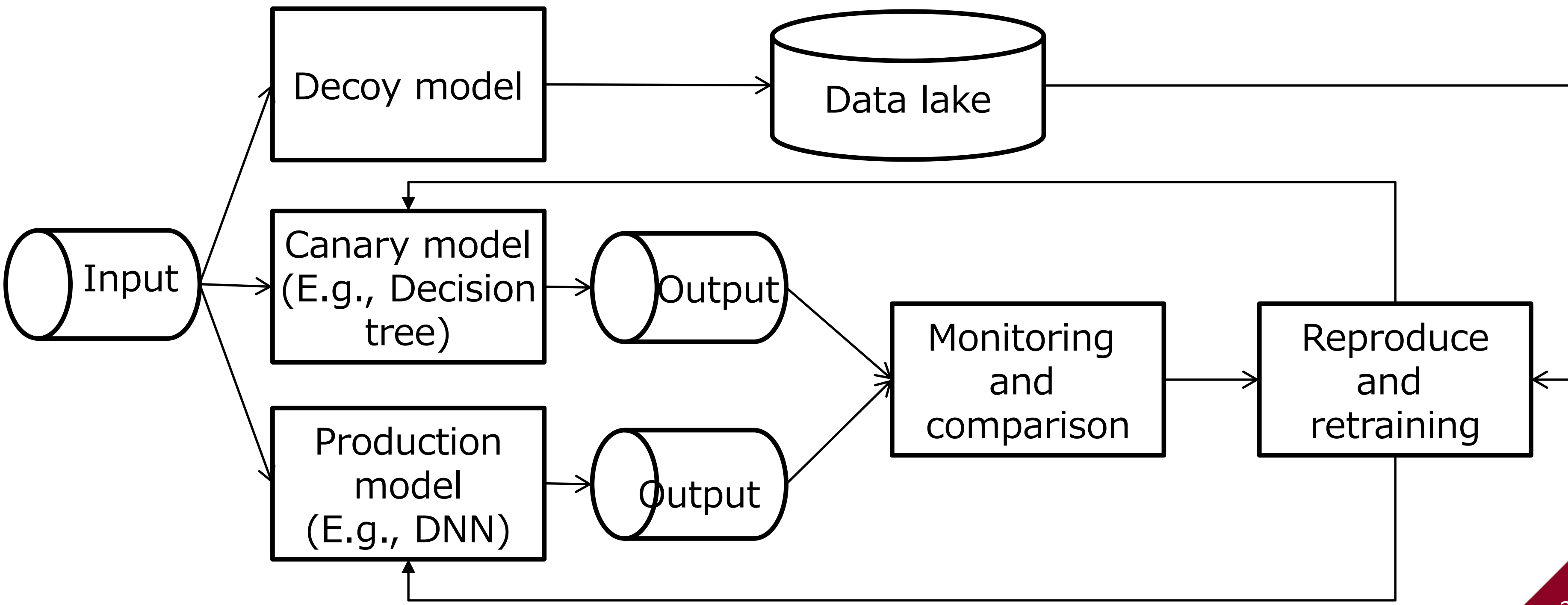
H. Washizaki, et al. Software Engineering Patterns for Machine Learning Applications (SEP4MLA) – Part 2, PLoP'20
Z. Peng, et al., A First Look at the Integration of Machine Learning Models in Complex Autonomous Driving Systems, ESEC/FSE'20

Model operation patterns

Pattern	Problem	Solution
Parameter-Server Abstraction	For distributed learning, widely accepted abstractions are lacking …	Distribute both data and workloads over worker nodes, while the server nodes maintain globally shared parameters …
Data Flows Up, Model Flows Down	Standard ML approaches require centralizing the training data on one machine …	Enable mobile devices to collaboratively learn while keeping all the training data on the device as federated learning …
Secure Aggregation	The system needs to communicate and aggregate model updates in a secure and scalable way …	Encrypt data from each device and calculate totals and averages without individual examination …
Deployable Canary Model	A surrogate ML that approximates the behavior of best model must be built to provide explainability …	Run the explainable inference pipeline in parallel to monitor prediction differences …
ML Versioning	ML models and their different versions may change the behavior of the overall ML applications …	Record the ML model, dataset, and code to ensure a reproducible training and inference processes …

Deployable Canary Model

- **Problem:** A surrogate ML that approximates the behavior of the best ML model must be built to provide explainability.
- **Solution:** Run the explainable inference pipeline in parallel with the primary inference pipeline to monitor prediction differences.
- Known usage: Image-based anomaly detection at factory

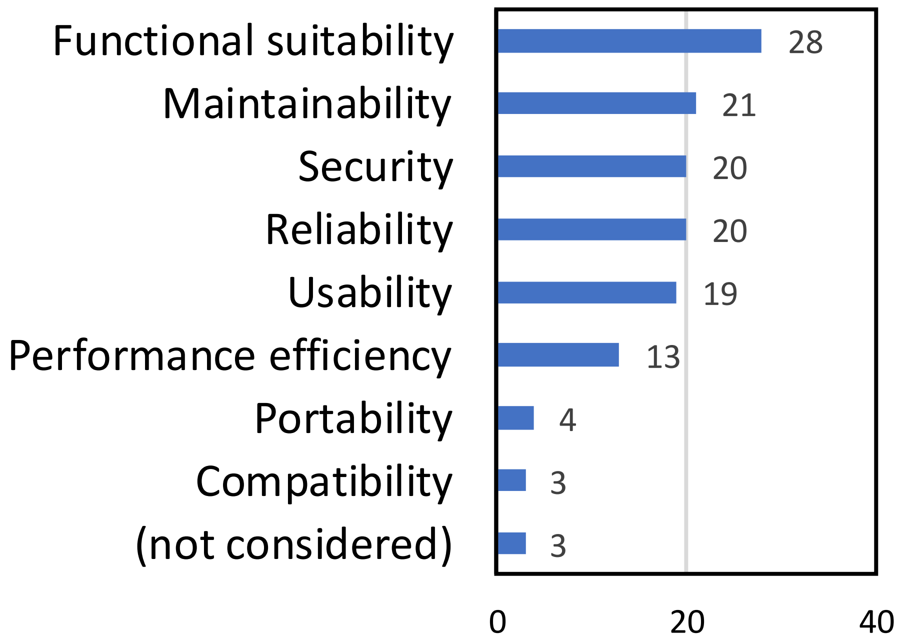


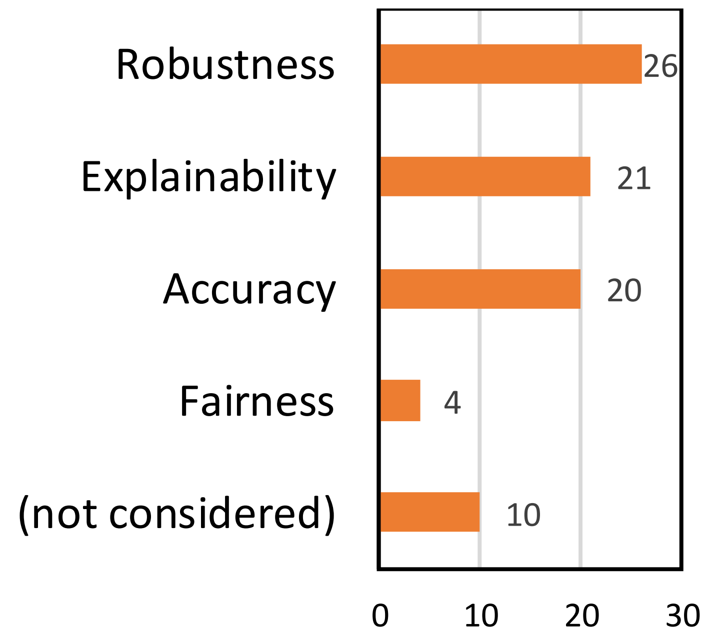
S. Ghanta et al., Interpretability and reproducibility in production machine learning applications, ICMLA 2018

Pattern	Performance	Compatibility	Reliability	Security	Maintainability	Portability	Robustness	Explainability	Accuracy
Different Workloads in Different Computing Environments	X				X				
Distinguish Business Logic from ML Models					X				
ML Gateway Routing Architecture		X			X				
Microservice Architecture for ML		X			X	X			
Lambda Architecture for ML	X		X						
Kappa Architecture for ML	X		X						
Data Lake for ML	X	X			X				
Separation of Concerns and Modularization of ML Components					X				
Encapsulate ML Models within Rule-based Safeguards			X						
Discard PoC Code					X				
Parameter-Server Abstraction	X		X						
Data Flows Up, Model Flows Down	X						X		X
Secure Aggregation				X			X		X
Deployable Canary Model			X					X	
ML Versioning					X		X		X

RQ3. Practitioners' insights on quality

- Surveyed 300+ developers, 46 answered in ML development
- **What product quality attributes considered?**
  - Maintainability, reliability, security, and usability
- **What model and prediction quality attributes?**
  - Robustness, accuracy, and explainability
- Maintainability, reliability, robustness and accuracy are well handled by ML patterns. **There are demands for having ML patterns addressing security, usability, and explainability, which are not handled well now.**

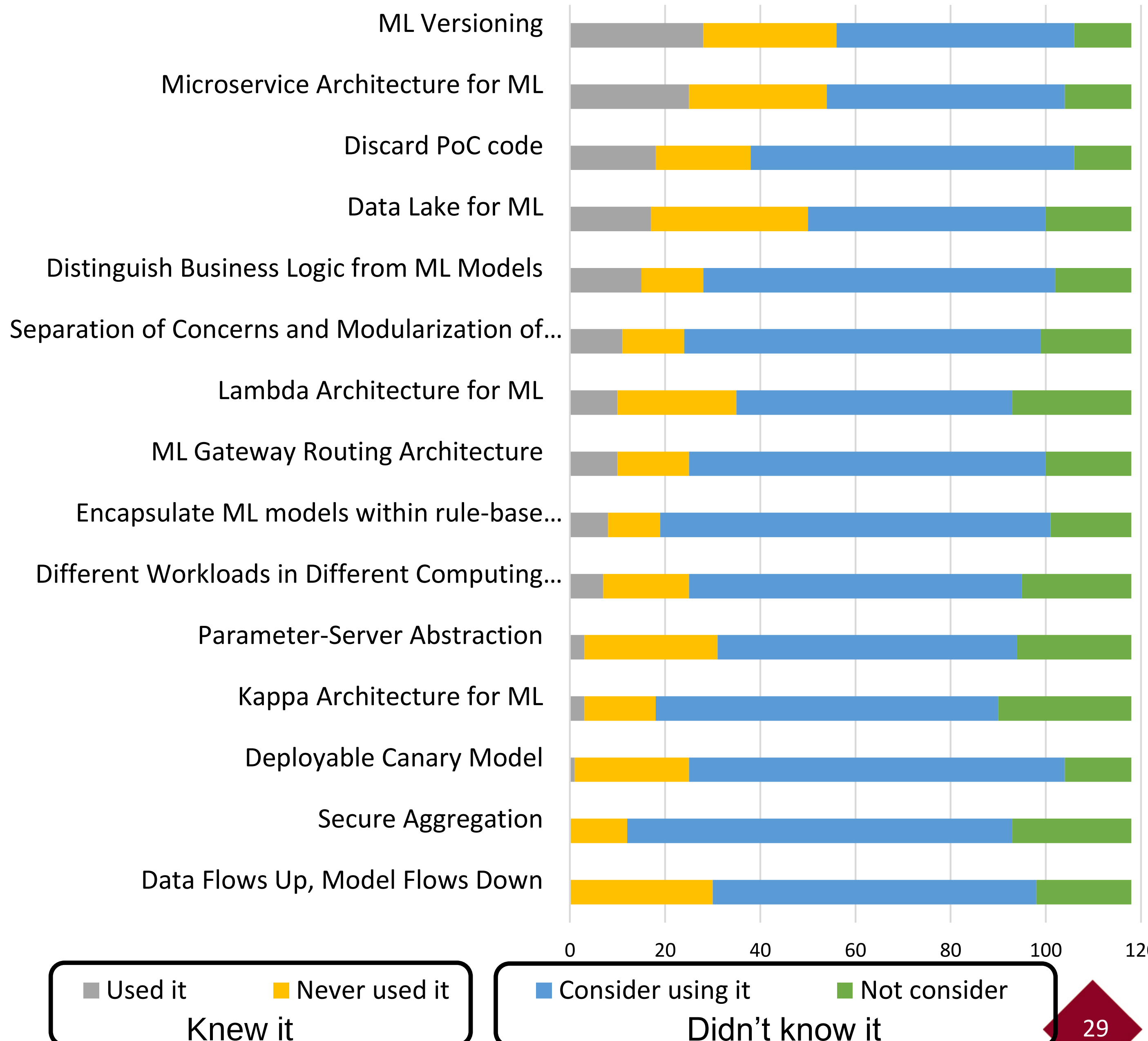




H. Washizaki, et al., Practitioners' insights on machine-learning software engineering design patterns: a preliminary study, ICSME 2020

Practitioners' insights on ML design patterns

- Surveyed 600+ developers, 118 answered
- **Have you ever referred to ML patterns?**
  - Major: **ML Versioning, Microservice Architecture for ML**
  - None: **Secure Aggregation, Data Flows Up** (aka. Federated Learning)



Practitioners' insights on ML design patterns

- **Have you ever referred to ML patterns?**
  - Developers were unfamiliar with most ML patterns, although there were several major patterns used by 20+% of the respondents.
  - For all patterns, most respondents indicated that they would consider using them in future designs.
  - Promoting existing ML patterns will increase their utilization
- **How do you solve and share design challenges of ML application systems?**
  - 37 (i.e., 31%) organized design patterns and past design results.
  - As respondents become more organized in their approach to design problems by reuse, the pattern usage ratio increased.
  - Development teams and organizations will reuse more ML patterns as they become more consistent in their reuse approach.

Design solution and reuse practice	#Respondents	#Patterns used	Pattern usage ratio
Lv3. Organizing, reusing patterns (and past results)	37	64	**11.5%**
Lv2. Reusing externally documented patterns	31	50	10.8%
Lv1. Resolving problems in an ad-hoc way	37	35	**6.3%**
Others	13	3	1.5%

Conclusion and future work

- Literature review of academic and gray literature
  - 15 SE patterns for ML applications extracted.
  - Patterns at https://eai-transfer.github.io/ml-design-pattern/en/
- Survey of practitioners' insights
  - Developers were unfamiliar with most ML patterns, although there were several major patterns used by 20% (such as **ML Versioning** and **Microservice Architecture for ML)**
- Identify ML patterns addressing specific quality attributes that are not handled well now
  - Security, usability, and explainability
- Future work
  - Investigate the impact of patterns on quality attributes of systems
  - Analyze relationships among patterns including related ones towards a pattern language
  - Integration into framework to handle from requirements on implementations and testing/debugging

Agenda

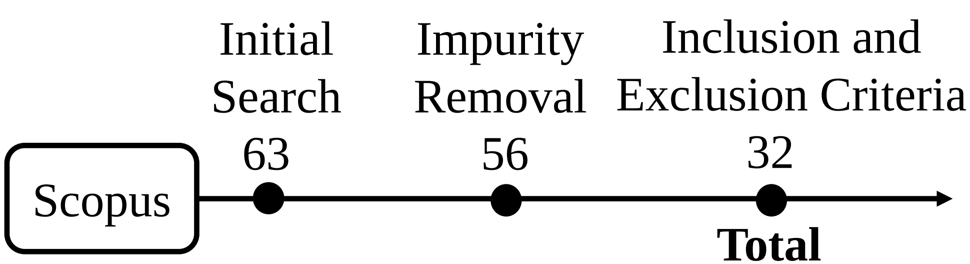
- Paradigm shifts in new software engineering
- Pattern language
- Machine learning patterns
- IoT patterns
- Security patterns

Executive summary

- IoT architecture and design patterns at different abstraction levels are not well classified and studied.
- **RQ1. How does academic literature address IoT architecture and design patterns?**
  - There are 32 academic papers related to IoT architecture and design patterns.
- **RQ2. Are all existing IoT architecture and design patterns really IoT patterns?**
  - Of the 143 extracted patterns, 57% are non-IoT patterns.
- **RQ3. Can IoT architecture and design patterns be classified?**
  - Patterns can be divided along three main characteristics: abstraction level, domain specificity, and quality attributes.
- **RQ4. What IoT architecture and design patterns exist?**
  - Many IoT patterns address interoperability, security, and maintainability.
  - Many IoT architecture patterns are domain-specific.

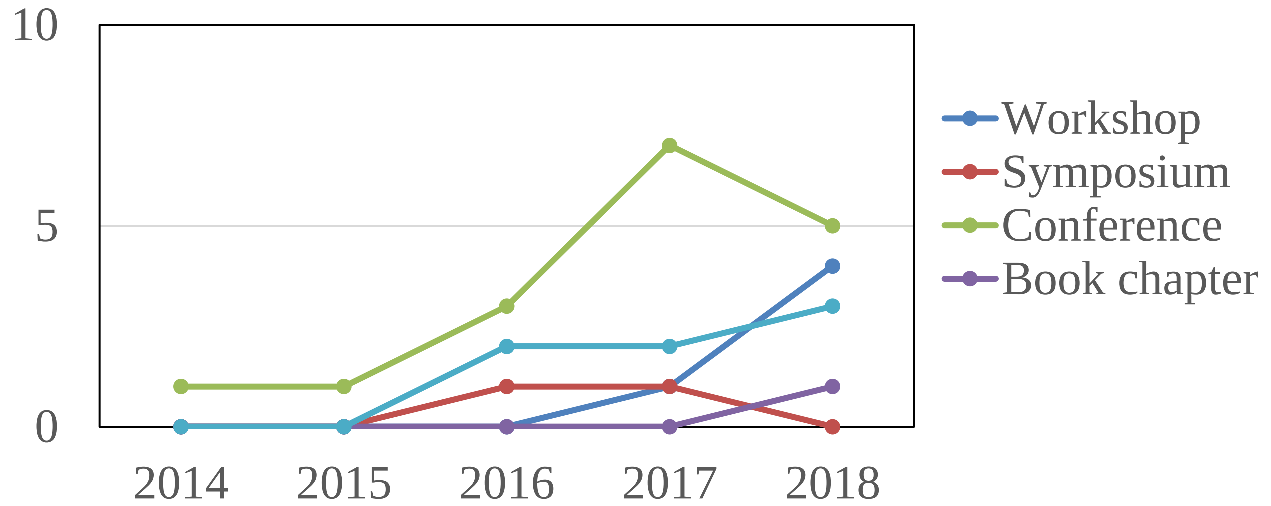
Systematic literature review (SLR)

- Initial Search: **63** papers **2014–2018 in Scopus**
  - “IoT” AND (“design pattern” OR “architecture pattern”)
- Impurity Removal: 56
- Inclusion and Exclusion Criteria: **32**
  - Inclusion: Addressing patterns for designing IoT systems and software, and papers written in English
- Data Extraction
  - Publication title, publication year, publication venue
  - Types of patterns proposed or used, pattern names
  - Domain names in the case of Specific IoT patterns
  - Quality attributes addressed



RQ1. How does academic literature address IoT architecture and design patterns?

- 32 academic papers related to IoT architecture and design patterns
  - Most are conference papers followed by journal publications.
- The high number of conference papers indicates that the **entire topic of IoT architecture and design patterns is in its early stage**
- But the presence of journal articles suggests that **some types of IoT patterns are maturing**.

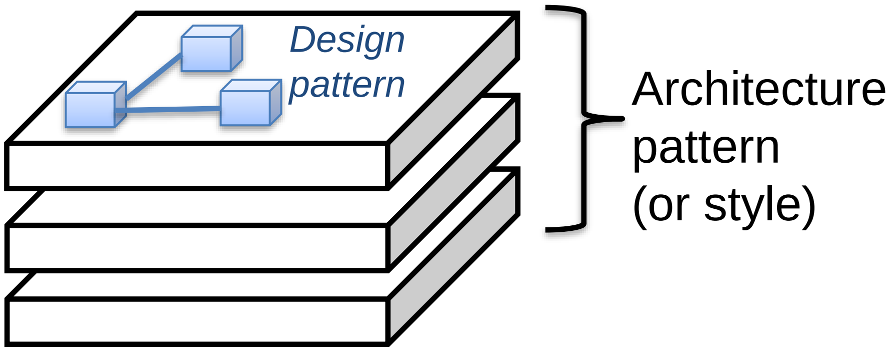


RQ2. Are all existing IoT architecture and design patterns really IoT patterns?

- 143 patterns mentioned in 32 papers
- **82 general (non-IoT) patterns**
  - Incl. 11 non-IoT patterns appeared in multiple papers: Publish-Subscribe, Client-Server, Peer-to-Peer, REST, SOA, RBAC, MVC, Reflection, Blockchain, Strategy and Observer
  - 14 papers used such **non-IoT patterns only**.
  - **IoT systems and software are often designed via conventional architecture and design patterns.**
- 61 IoT patterns in 18 papers

RQ3. Can IoT architecture and design patterns be classified?

- **Abstraction level**
  - High: Architecture styles
  - Middle: Architecture patterns
  - Low: Design patterns

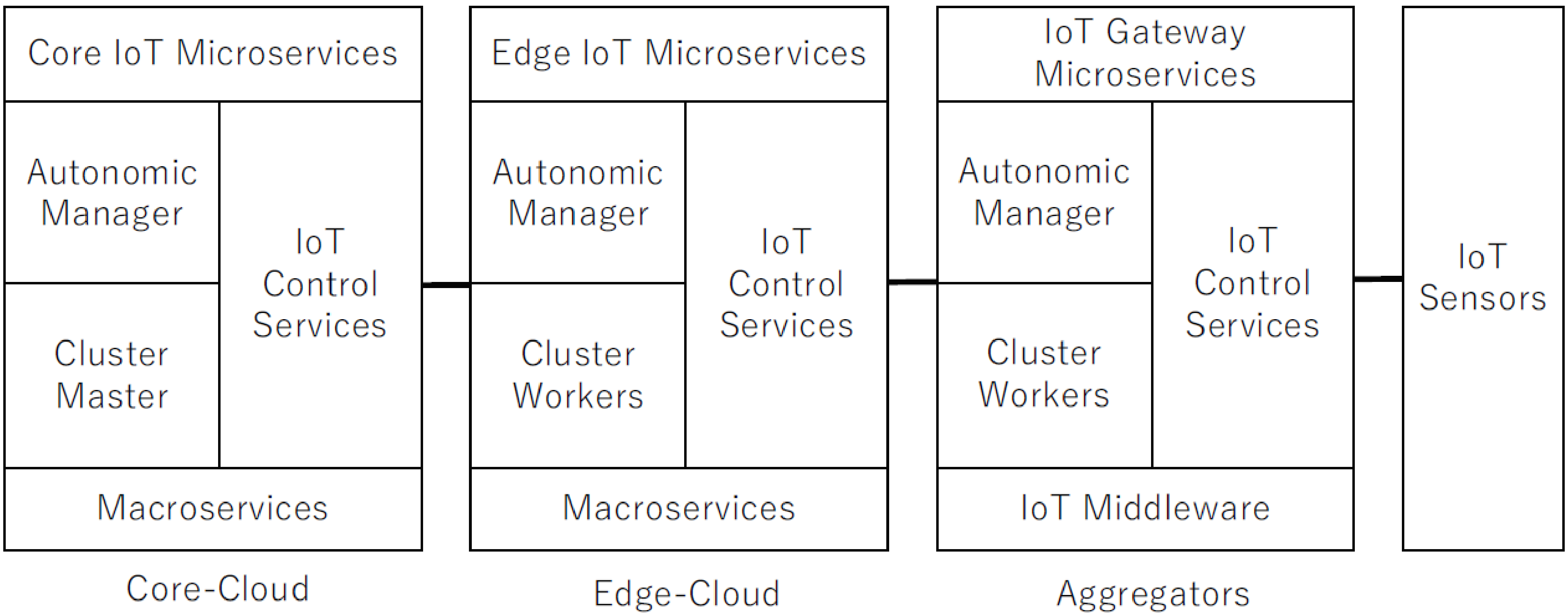


- **Domain specificity**
  - Any: General architecture/design patterns those can be adopted to design IoT systems and software
  - General IoT: Applicable to any IoT systems and software.
  - Specific IoT: Addressing specific problem/technical domains
- **Quality characteristic**
  - All quality characteristics except for functional suitability in ISO/IEC 25010

    \+ Emerging characteristics common in IoT such as scalability and privacy

E.g.: Layered architecture for IoT applications

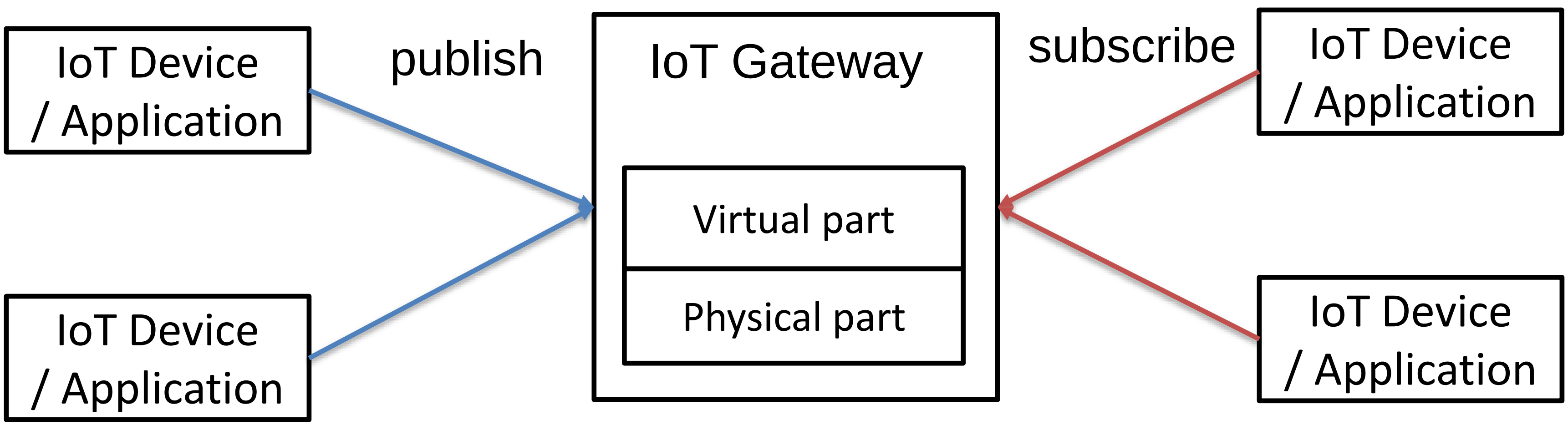
- IoT platform providing resource virtualization using **lightweight virtualization for multi-layer** applications



H. Khazaei, H. Bannazadeh, and A. Leon-Garcia, "SAVI-IoT: A self-managing containerized IoT platform," in 5th IEEE International Conference on Future Internet of Things and Cloud, FiCloud 2017,

E.g.: IoT Gateway Event Subscription

- Employ a **subscription** mechanism into the IoT gateway
- Allowing **asynchronous and mutual transmissions** of data obtained by sensors at the destination and the message between artifacts



R. Tkaczyk, K. Wasielewska, M. Ganzha, M. Paprzycki, W. Pawlowski, P. Szmeja, and G. Fortino, "Cataloging design patterns for internet of things artifact integration," in 2018 IEEE International Conference on Communications Workshops, ICC Workshops 2018

RQ4. What IoT architecture and design patterns exist?

- IoT patterns are **not recognized by different author groups**
  - Only two patterns mentioned in multiple papers
  - Pattern authors are encouraged to carefully check existing IoT patterns
- Combinations of abstraction level and domain specificity
  - Most of IoT **design patterns are applicable to any** domain
  - Many IoT **architecture patterns exist for specific** domains
  - **Unique nature of IoT adoption in specific domains appears at the architecture level**
- Major quality attributes: **Interoperability, security and maintainability**

	Domain specificity			
Type	Non-IoT	General IoT	Domain-specific IoT	Total
Architecture style	22	2	1	25
Architecture pattern	7	1	**15**	23
Design pattern	53	**38**	4	95
Total	82	41	20	143

Conclusions and future work

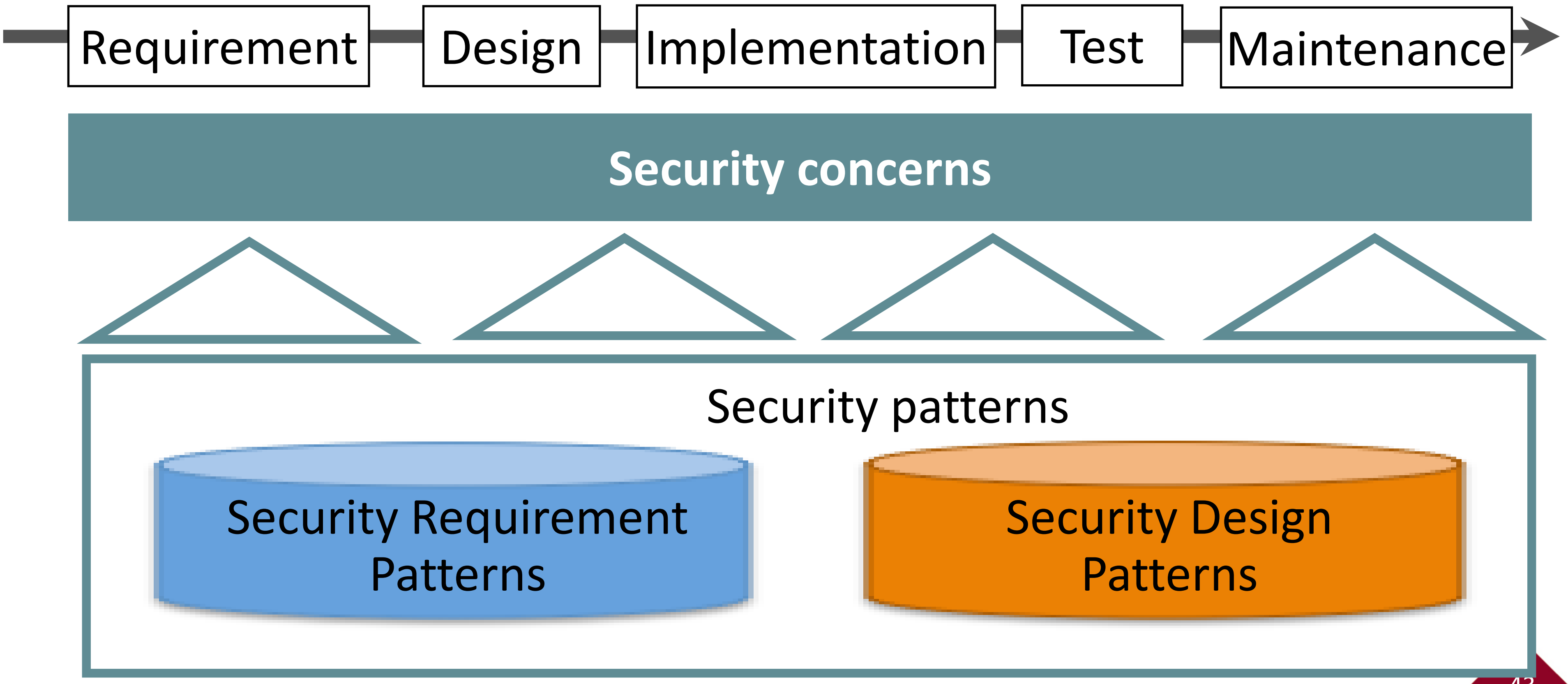
- We surveyed 143 patterns mentioned in 32 papers
  - Most of IoT design patterns are applicable to any domain but many **IoT architecture patterns exist for specific domains**.
  - Many IoT patterns address **interoperability, security and maintainability**. Other quality attributes remain to be researched.
- Further directions
  - We opened the classification results to the public and call for comments at our Website. http://www.washi.cs.waseda.ac.jp/iot-patterns/
  - Only Scopus for SLR. We plan to additionally use other databases.
  - Needs to analyze relationships among IoT patterns towards a pattern language

Agenda

- Paradigm shifts in new software engineering
- Pattern language
- Machine learning patterns
- IoT patterns
- Security patterns

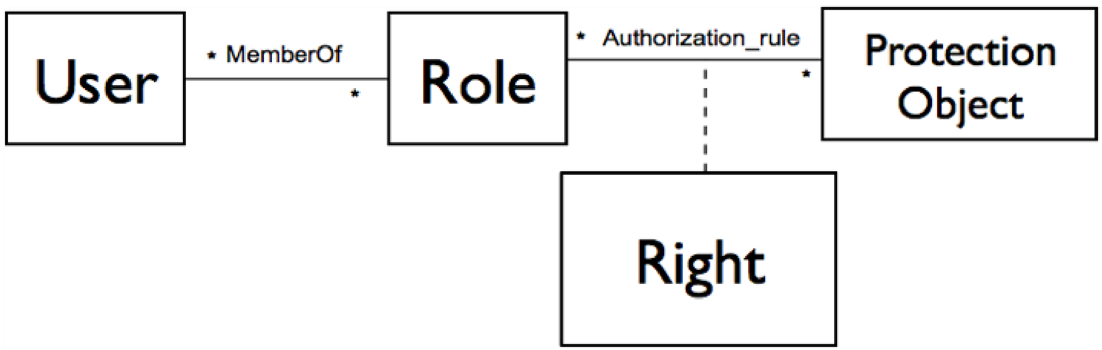
Security concerns must be addressed at any phase

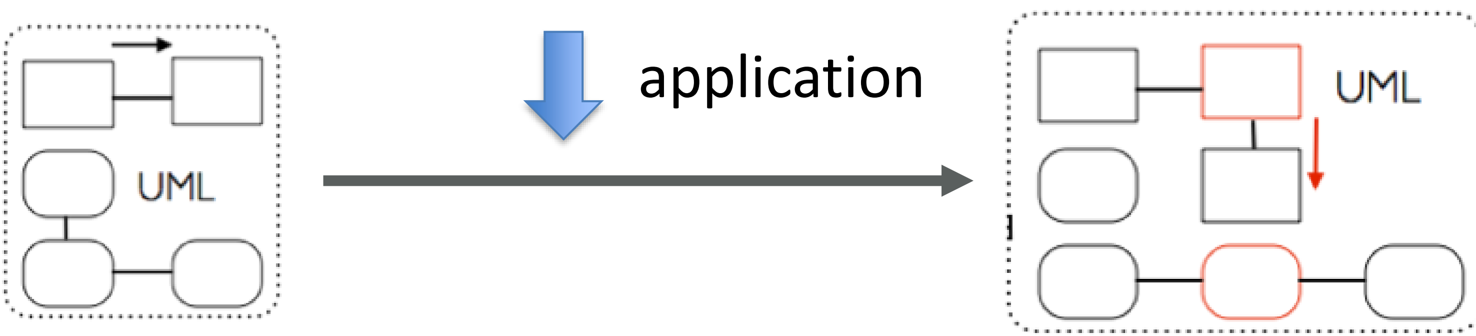
- Patterns are recurrent problems and solutions under specific contexts from requirements to maintenance



Example of security pattern

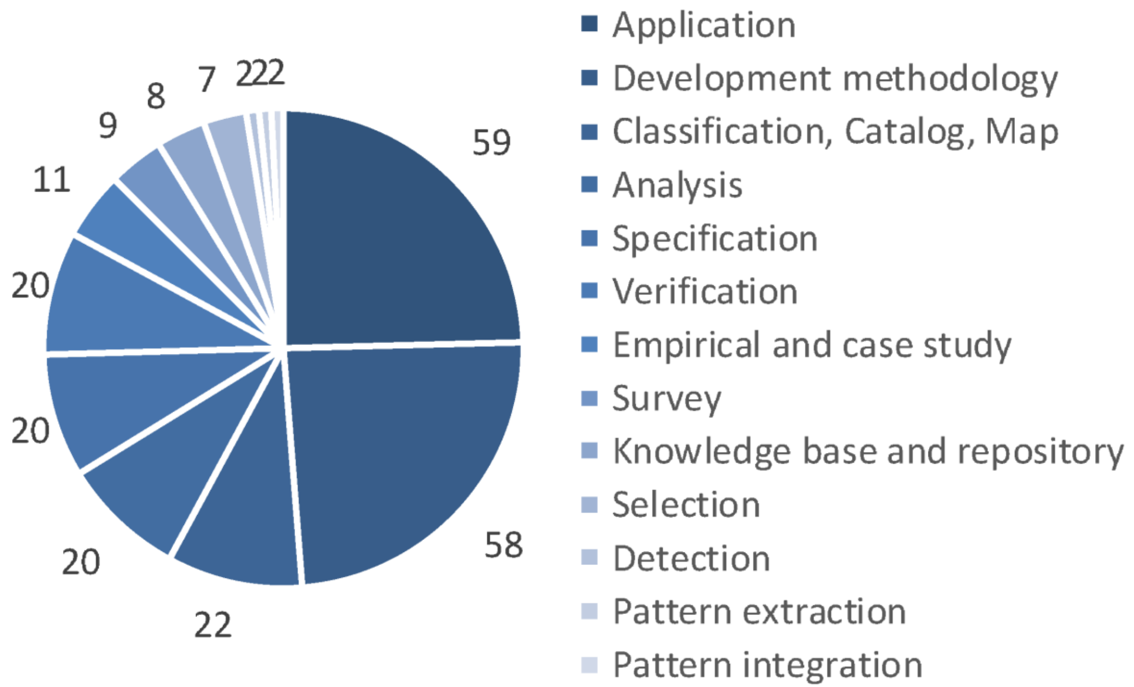
- Name: ***Role-based access control (RBAC)***
- Problem: How do we assign rights to people based on their functions or tasks?
- Solution: Assign users to roles and give rights to these roles so they can perform their tasks.
- Related patterns: ***Authorization,*** . . .





Systematic Literature Review of Security Pattern Research

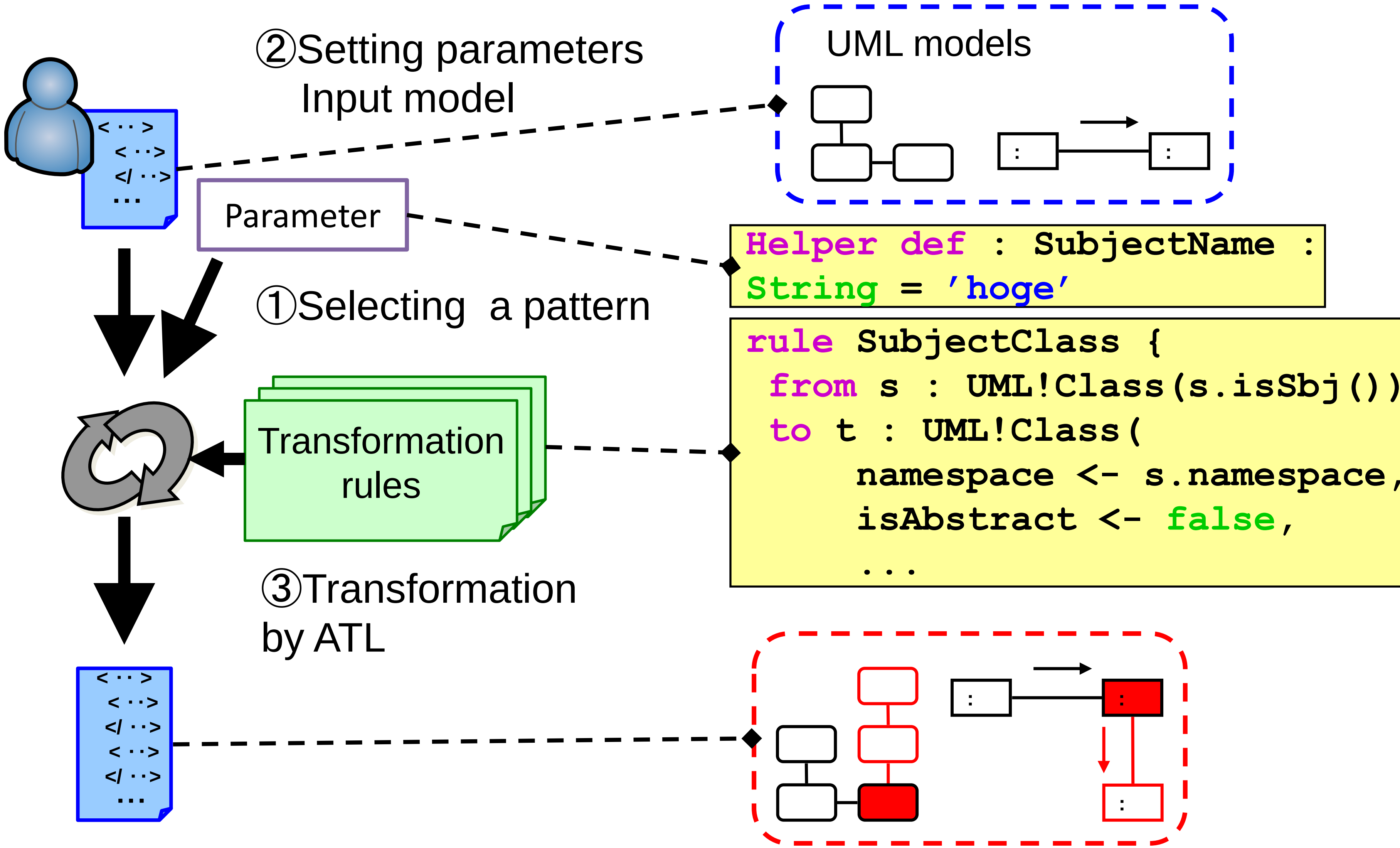
- We categorize and analyze 240 papers to clarify state-of-the-art and future directions of security pattern research in terms of 13 facets including topics and security characteristics.
- E.g., breakdown of research topics



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.

Model-driven security pattern application

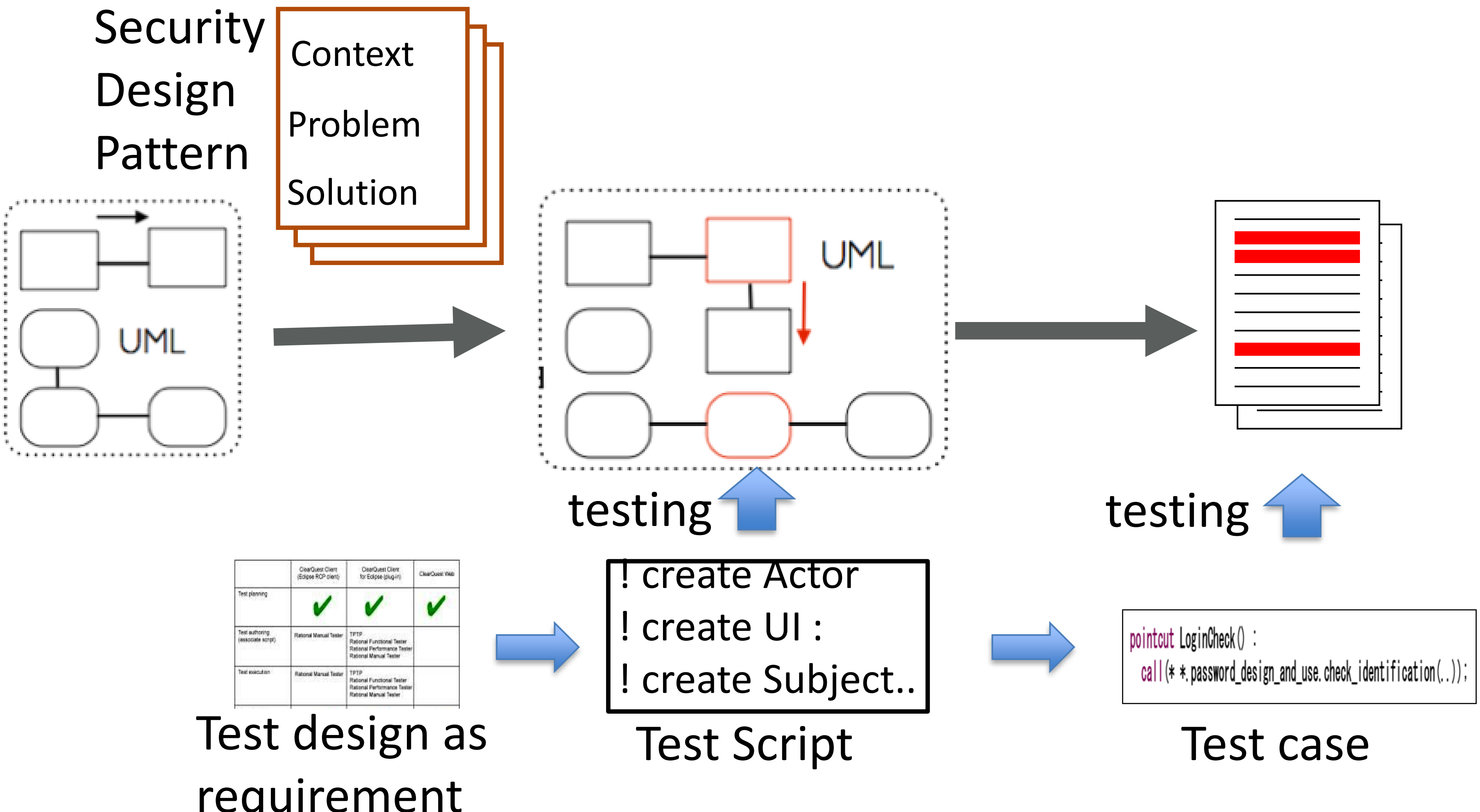
[PLoP’10]



[PLoP’10] Model-Driven Security Patterns Application and Validation,” 17th Conference on Pattern Languages of Programs

TESEM: Test Driven Secure Modeling Tool

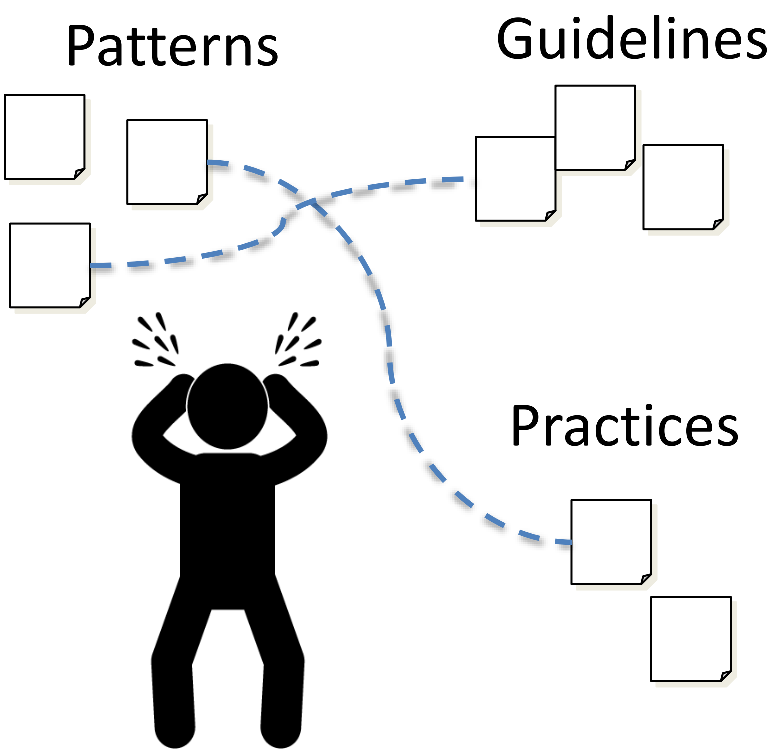
[ARES'13][ARES'13][IJSSE'14][ICST'15][Information'16]

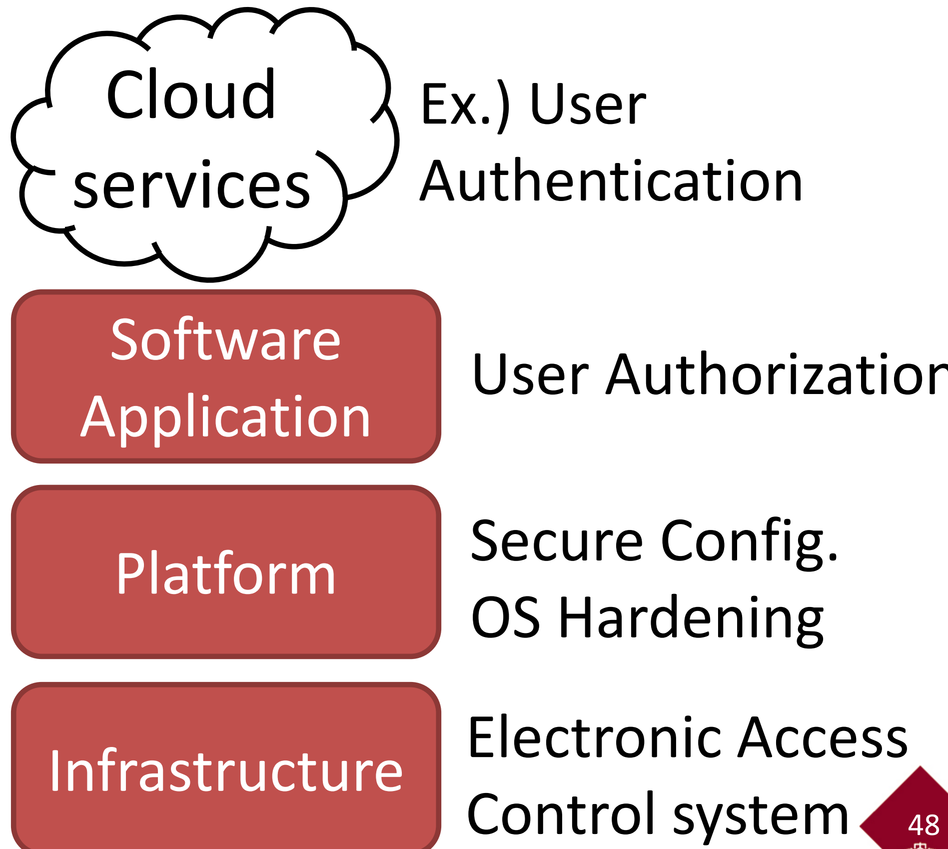


[ARES'13] Validating Security Design Pattern Applications Using Model Testing, Int'l Conf. Availability, Reliability and Security
[ARES'14] Verification of Implementing Security Design Patterns Using a Test Template, Conf. Availability, Reliability and Security
[IJSSE'14] Validating Security Design Pattern 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)

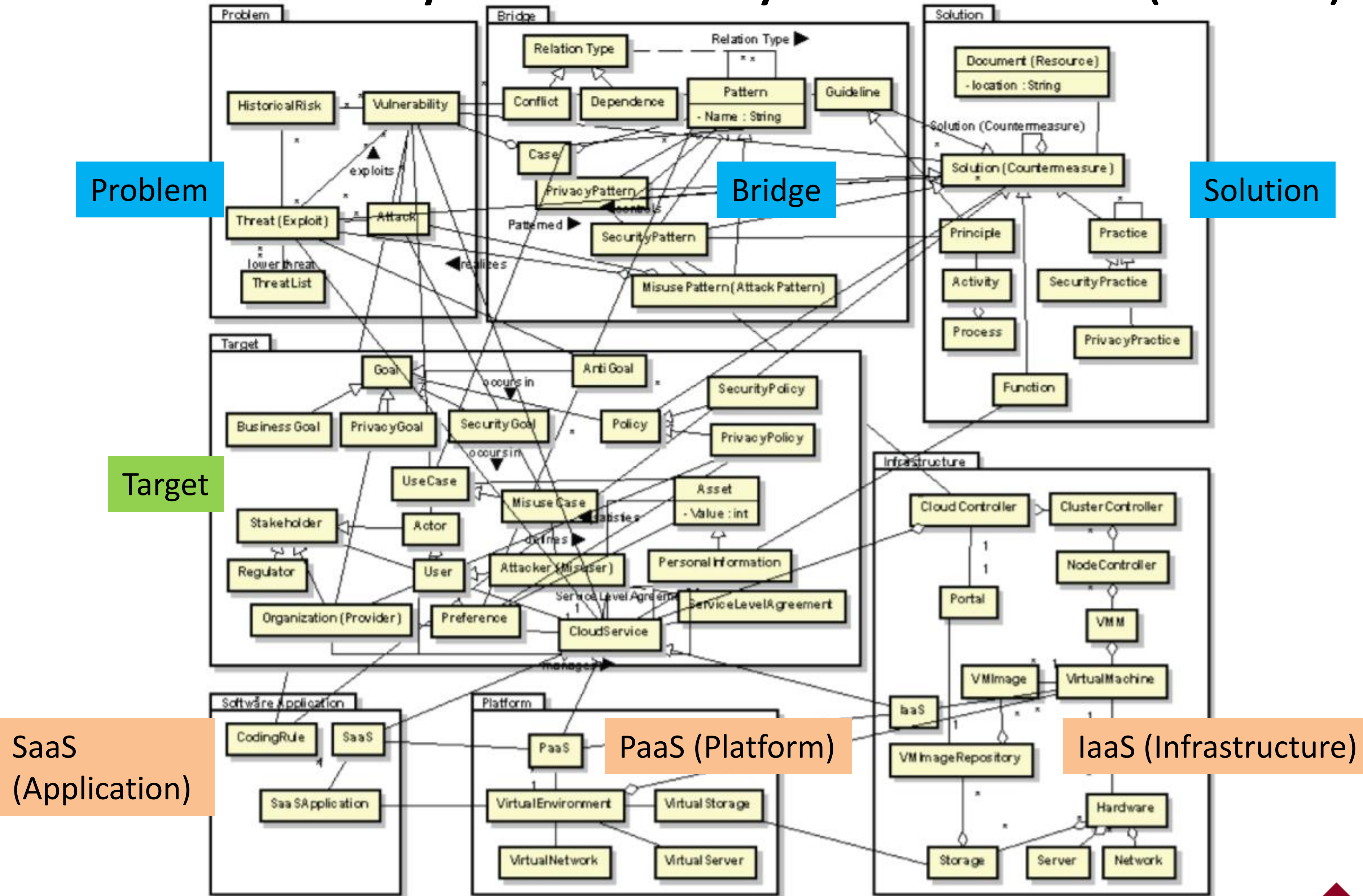
Challenges in cloud security and privacy (S&P)

- How to consistently utilize diverse S&P knowledge?
  $\Rightarrow$ Metamodel
- How to consider S&P over different layers?
  $\Rightarrow$ Layered metamodel





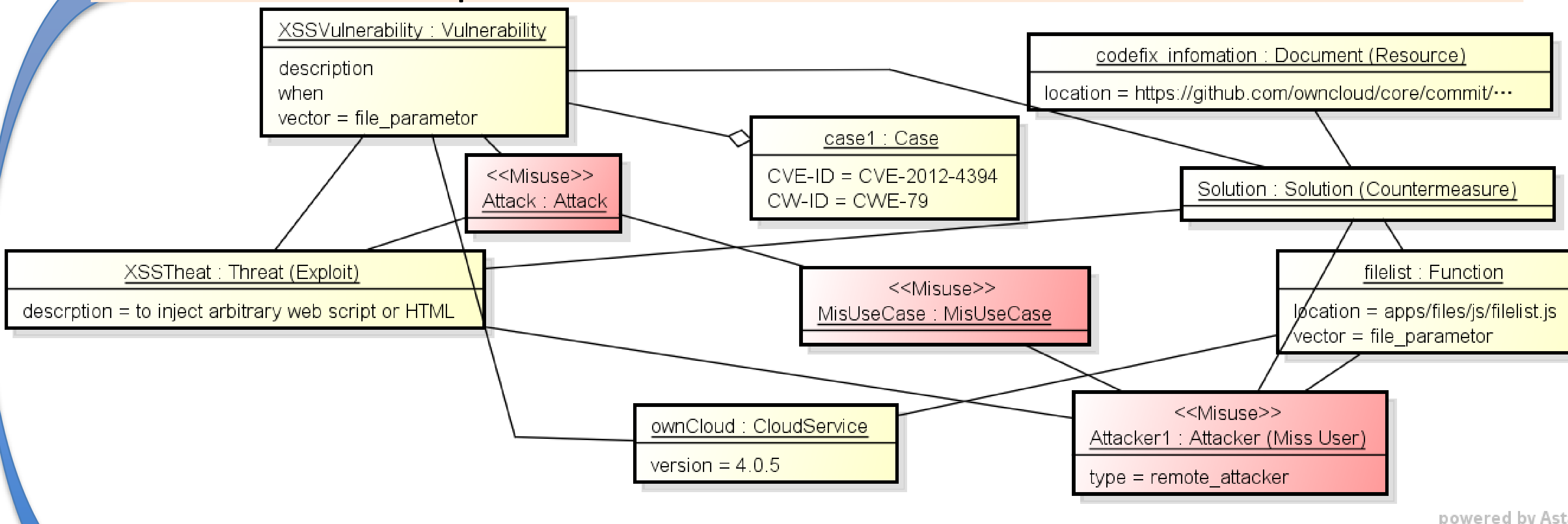
Cloud Security and Privacy Metamodel (CSPM)



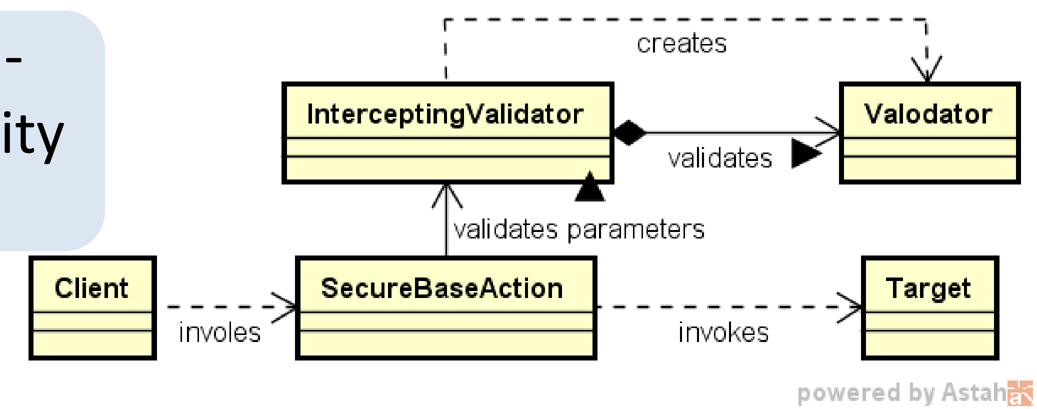
Tian Xia, Hironori Washizaki, Yoshiaki Fukazawa, Haruhiko Kaiya, Shinpei Ogata, Eduardo B. Fernandez, Takehisa Kato, Hideyuki Kanuka, Takao Okubo, Nobukazu Yoshioka and Atsuo Hazeyama, “CSPM: Metamodel for Handling Security and Privacy Knowledge in Cloud Service Development,” International Journal of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

Modeling vulnerability and security pattern

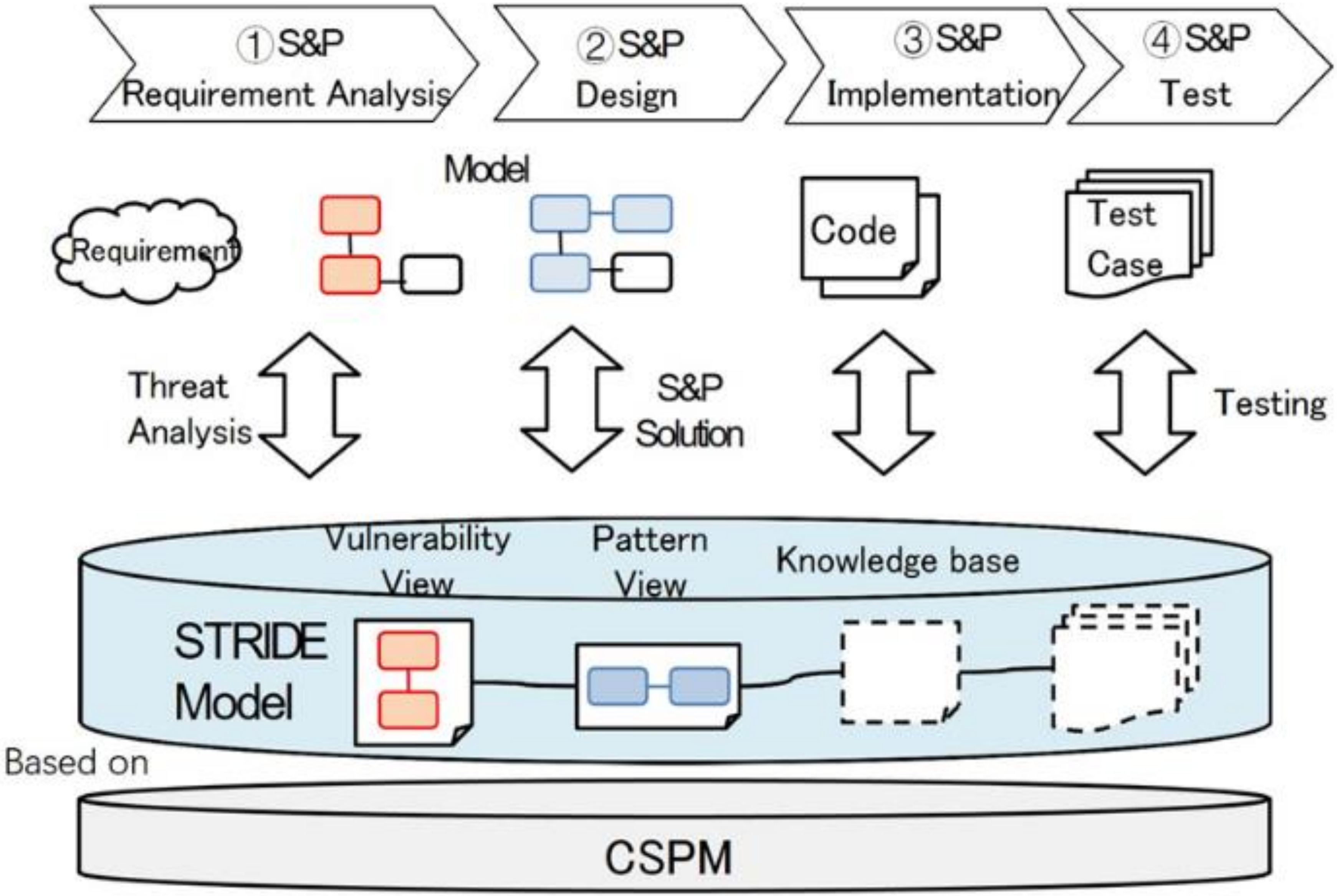
Common Vulnerabilities and Exposures: CVE-2012-4394 Cross-site scripting (XSS) vulnerability in apps/files/js/filelist.js in own Cloud before 4.0.5 allows remote attackers to inject arbitrary web script or HTML via the file parameter.



Validator for data-injection vulnerability such as XSS



Security and privacy development process



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	⇨	IoT and security education program

Summary

- There are paradigm shifts in “new” software engineering.
  - ML and IoT are key enablers of digital transformations.
  - Security must be a critical cross-cutting concern in ML and IoT software.
- New software engineering needs patterns and pattern languages.
  - Bridge between abstract paradigms and concrete cases/tools
  - Common language among stakeholders
- Future
  - Classify and relate patterns across over different disciplines such as ML and IoT
  - Build pattern languages
  - Open expanded community