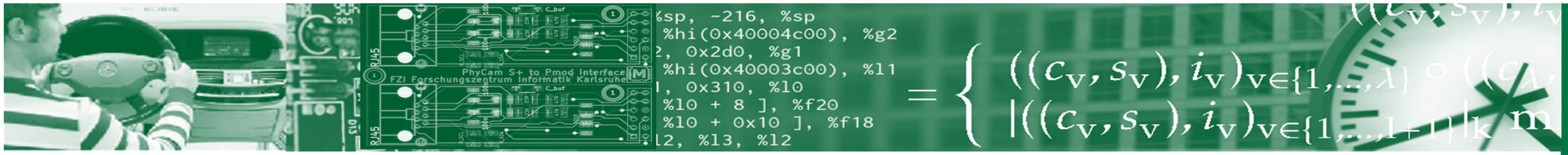


Security analysis of embedded systems using virtual prototyping

Presenter: Yasamin Mahmoodi, mahmoodi@fzi.de

Authors: Yasamin Mahmoodi, Sebastian Reiter, Alexander Viehl, Oliver Bringmann

FZI Research Center for Information Technology

Yasamin Mahmoodi

- PhD candidate in Tübingen University, Germany since 2016 Works as scientific researcher in FZI
- (Forschungszentrum Informatik), Germany since 2016
- Research Thema: security analysis for embedded systems

Problem

Foundation

Connected embedded systems

- Advantages
  - Efficient resource utilization
  - Minimizing human effort
  - Saving time

- Problems
  - Security
  - Privacy
  - Complexity

Secure embedded systems

Challenges facing secure embedded system design

- Security as add-on feature to embedded systems
- Weakness elimination efficiency
- Design decision verification from security aspects
- Security processing gap
- Entangled structure of embedded systems

Proposed solution

- Security by design
- model-orientation
- Security mechanisms estimation
- Refinement process
- Covering hardware architecture and network structure

Proposed Solution

Proposed approach

- **Security by design**
- **Model-orientation**
- **Security mechanisms estimation**
- **Refinement process**
- **Covering hardware architecture and network structure**

Security by design

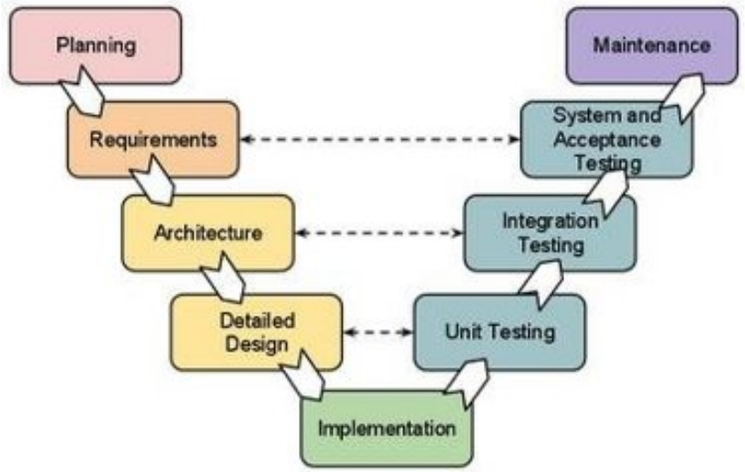
with the help of

virtual prototyping

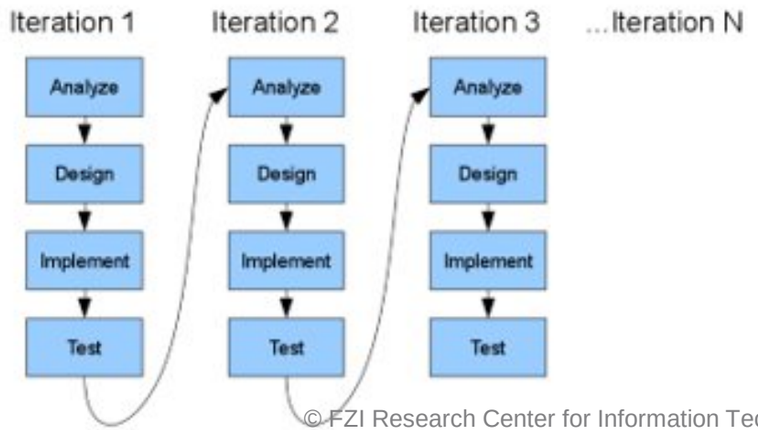
Security by design

SDLC (System Development Life Cycle) models:

- V-shaped model

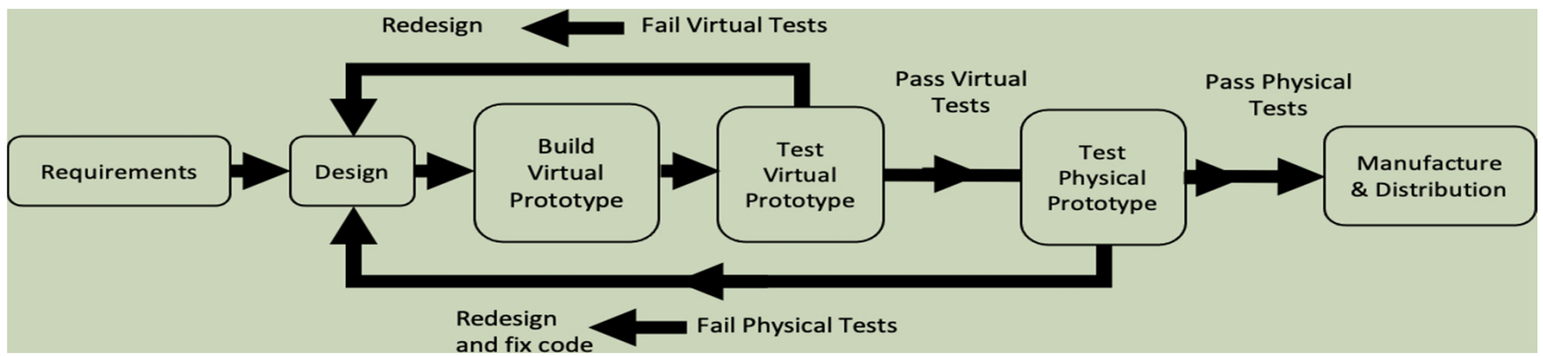


- Iterative model

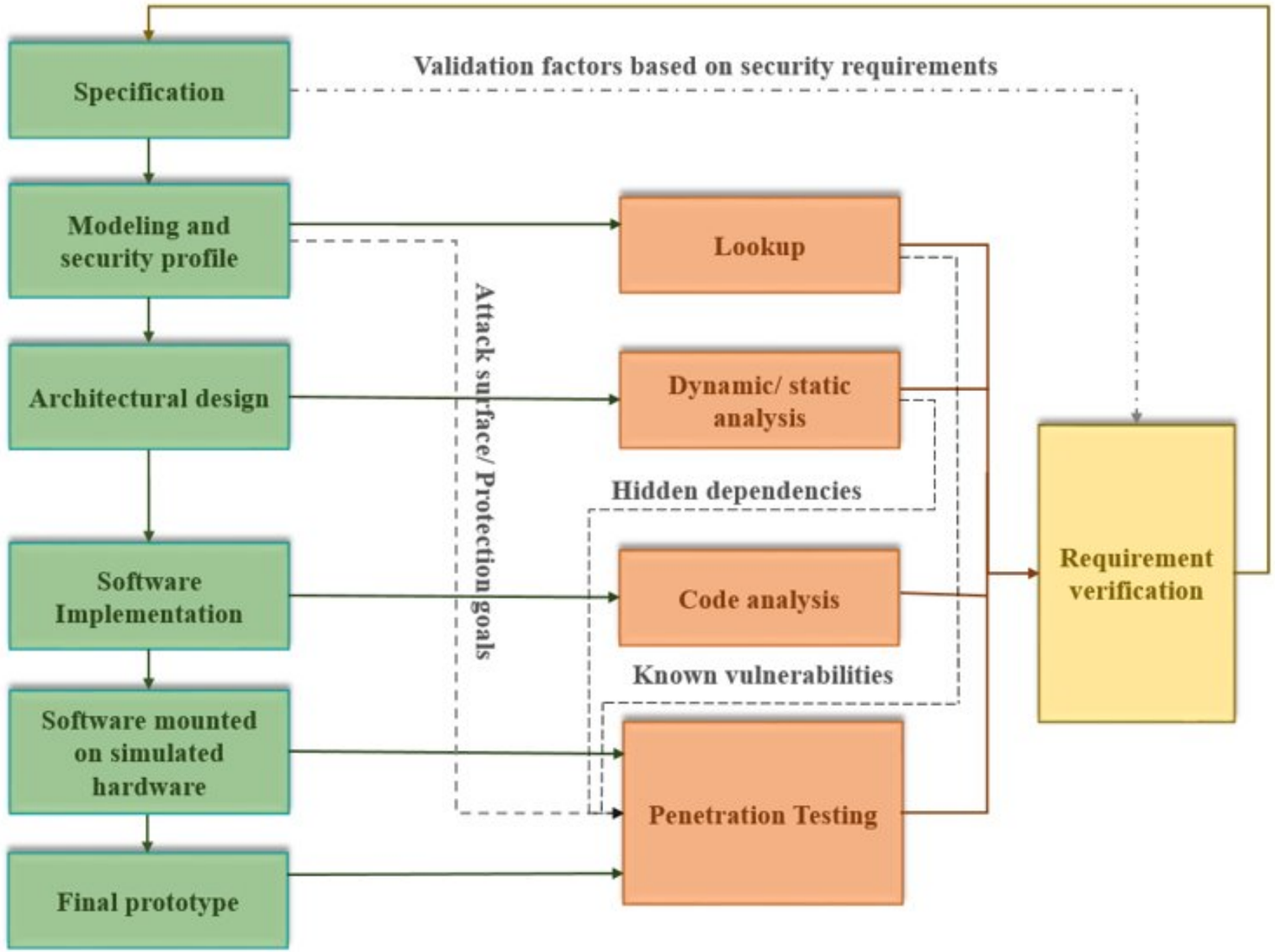


Virtual prototyping

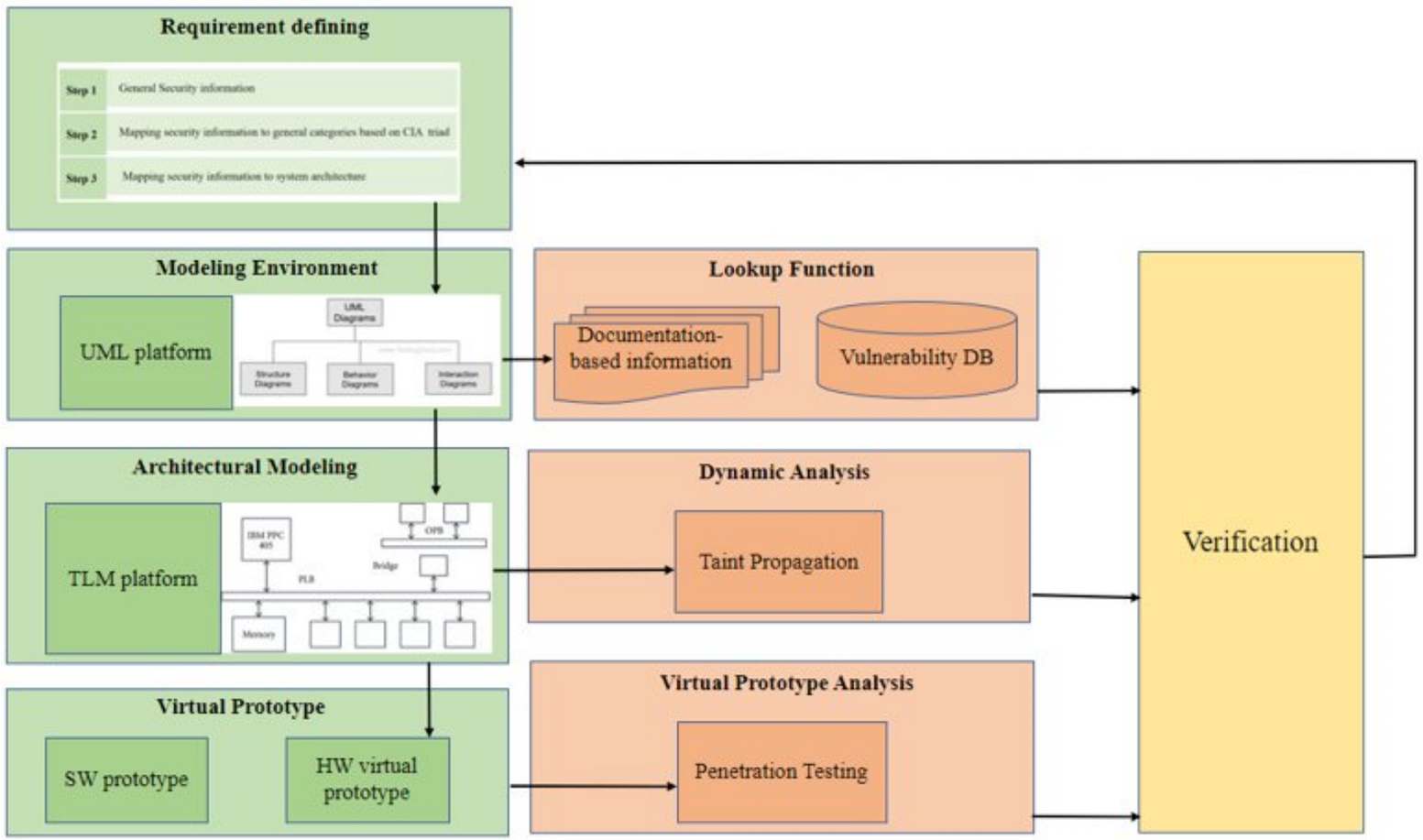
- Preliminary stage of physical prototype
- Completely or partial provision of subcomponents
- Executable model
- Widely used in product development and system analysis
- Simulation language : SystemC
  - Different abstraction level of SW/ HW
  - Modeling of SW application/ Digital and analog electronic components



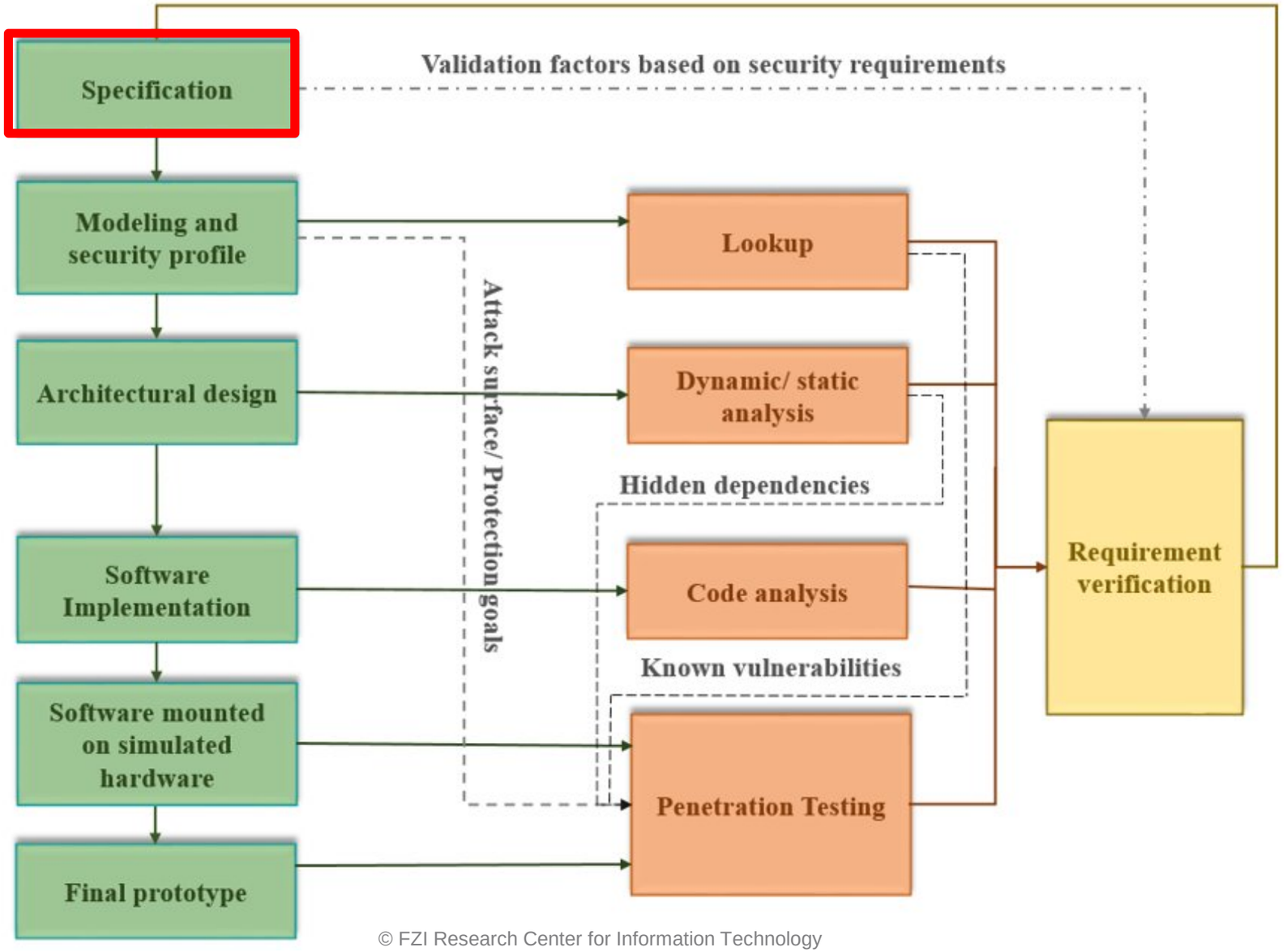
Proposed approach

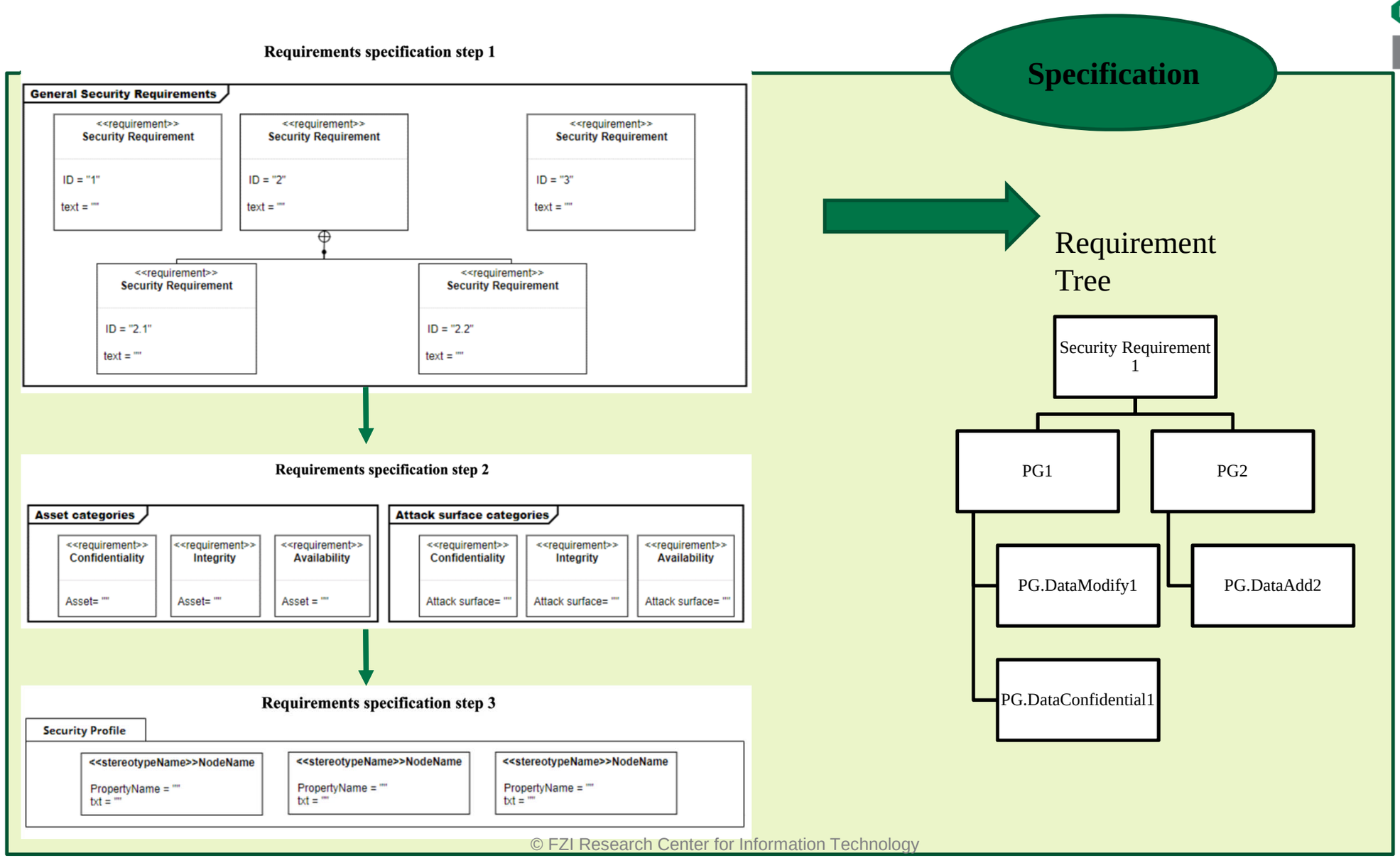


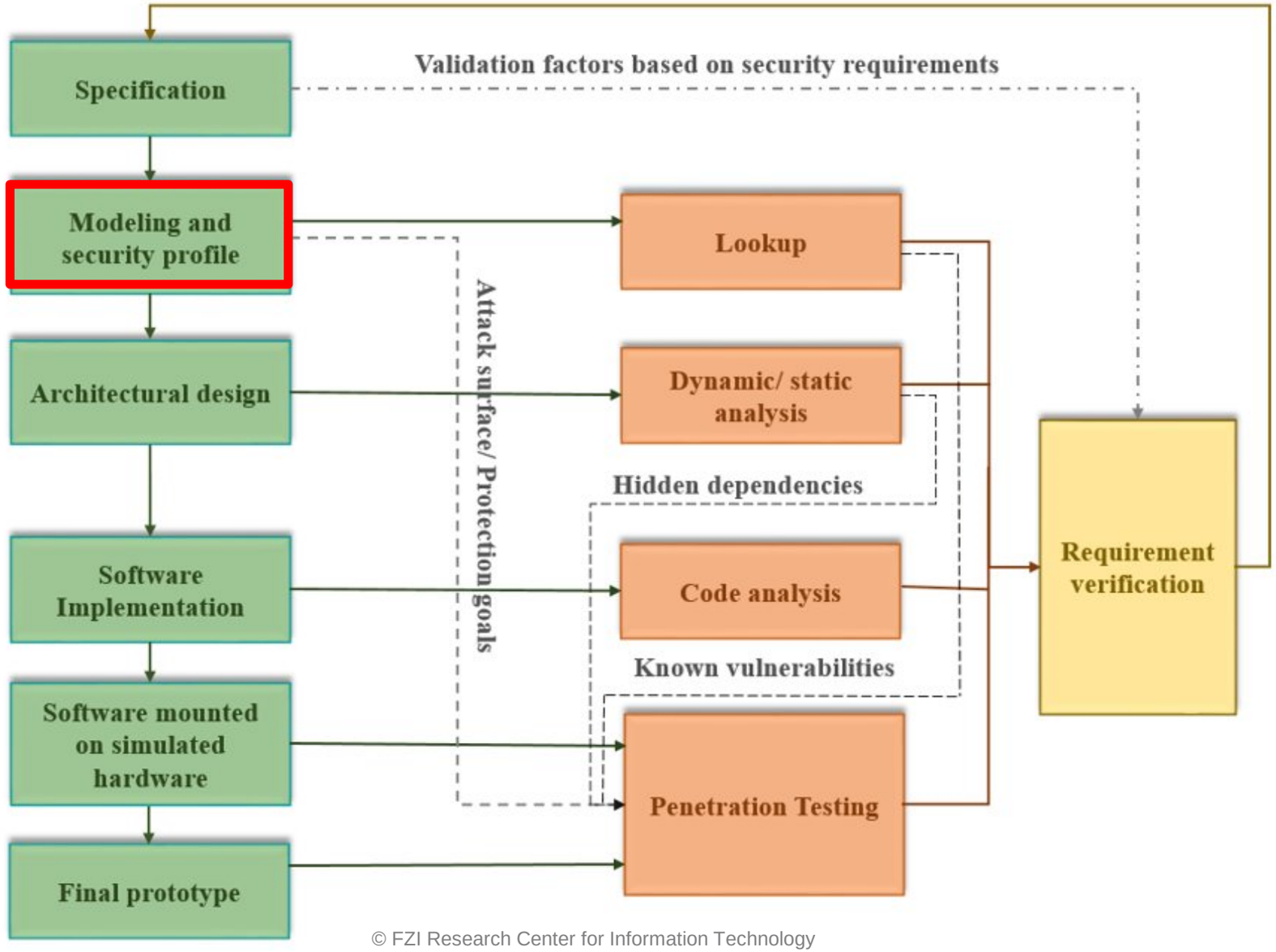
Implementing approach

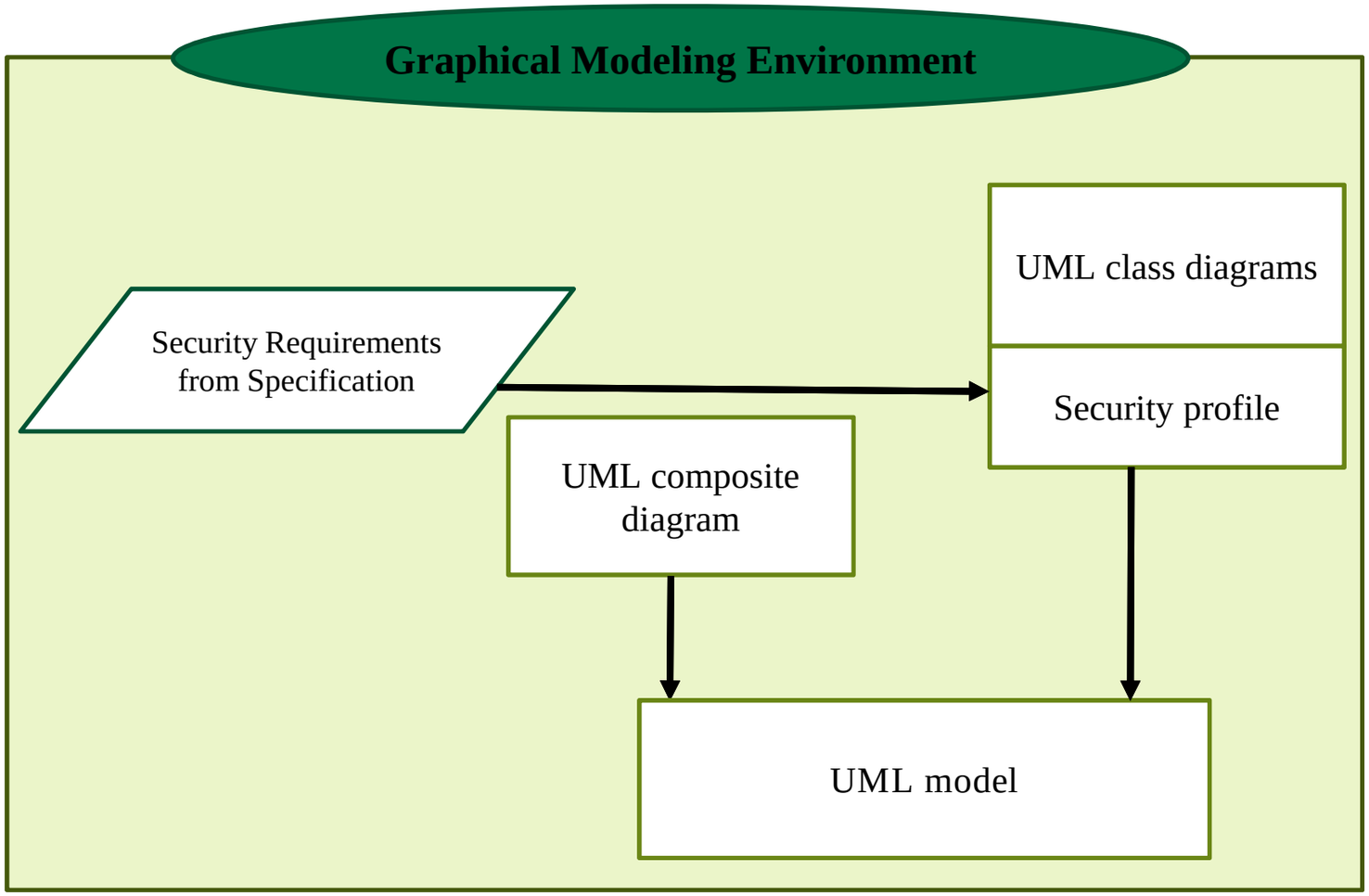


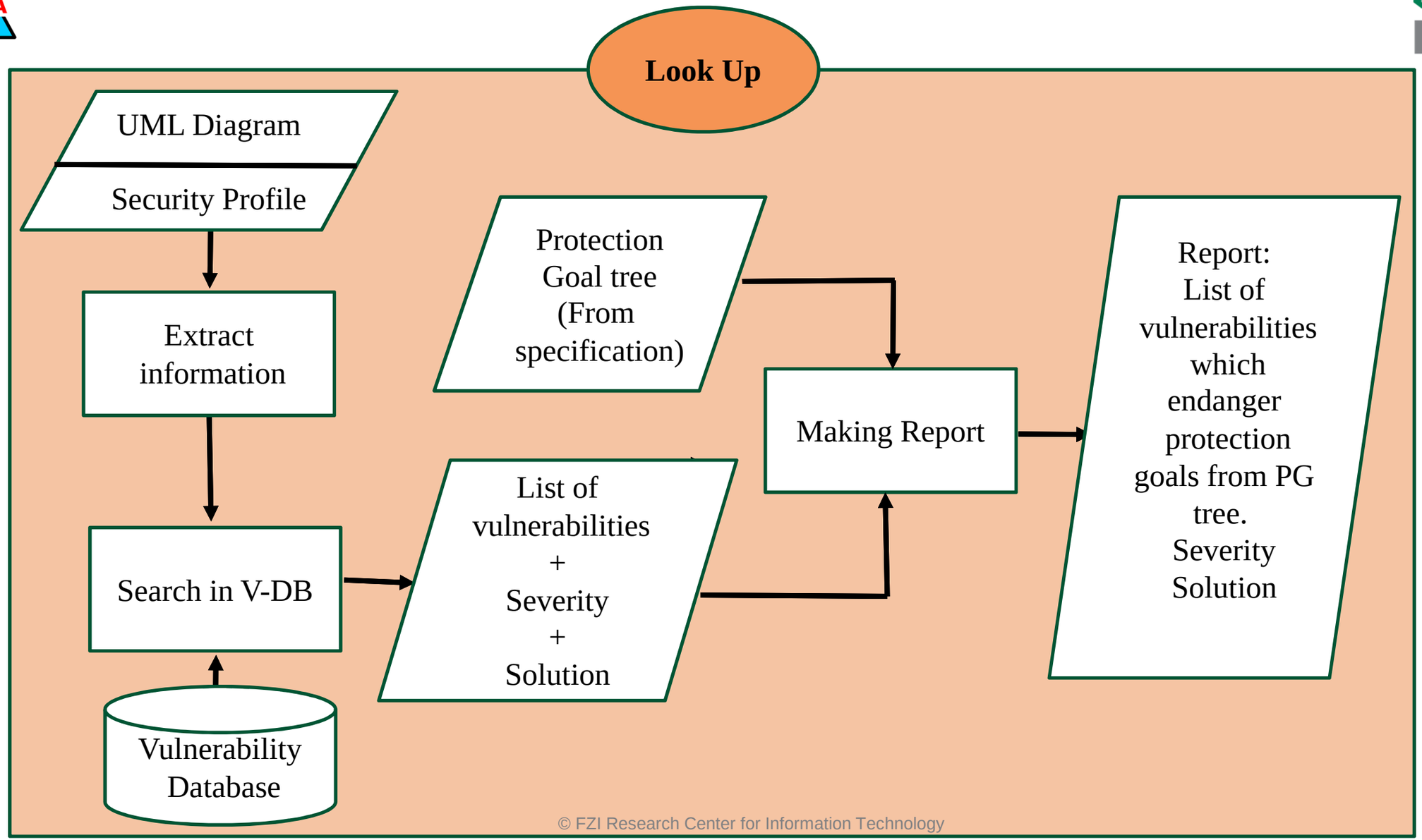
Implementation

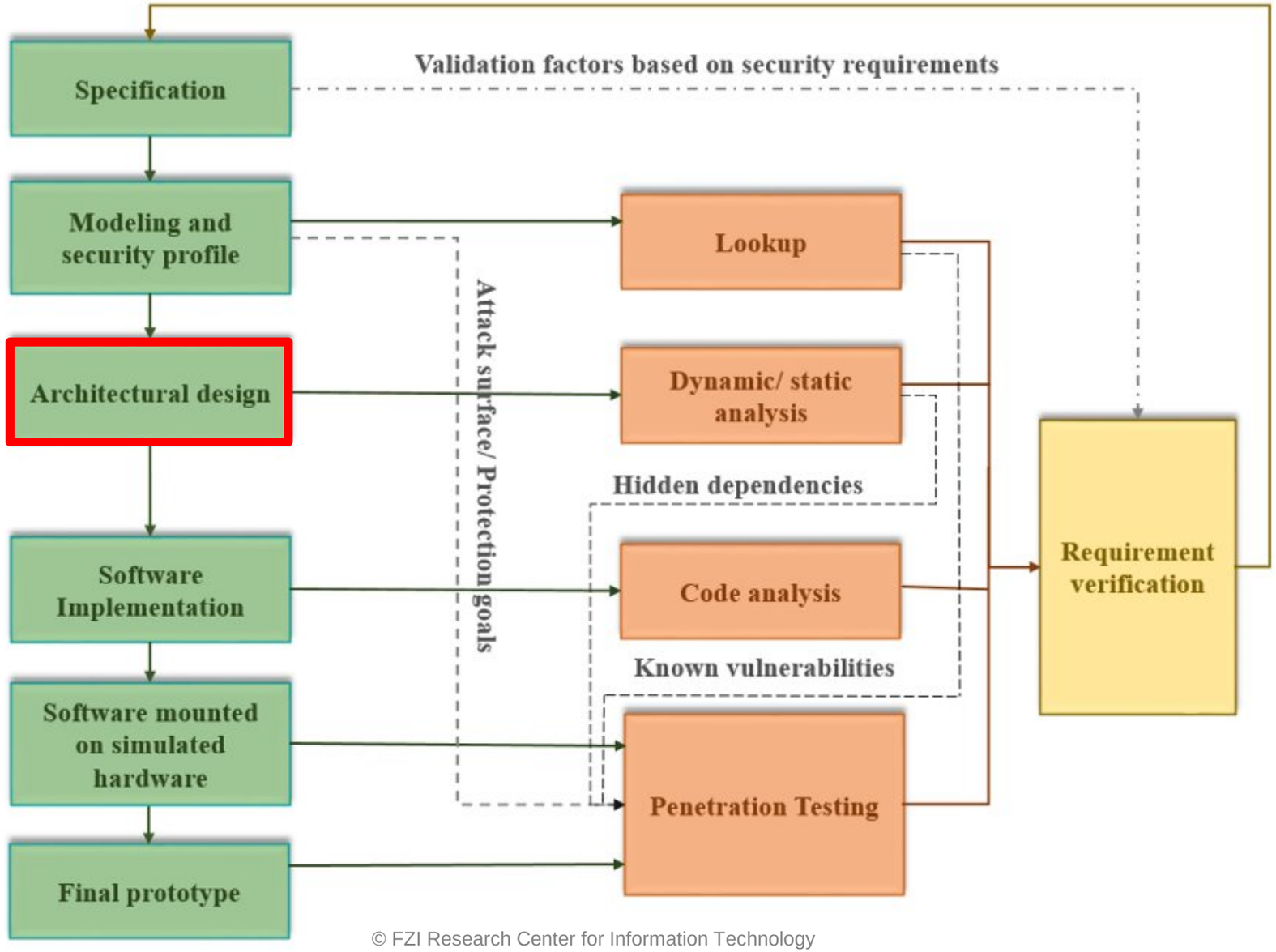


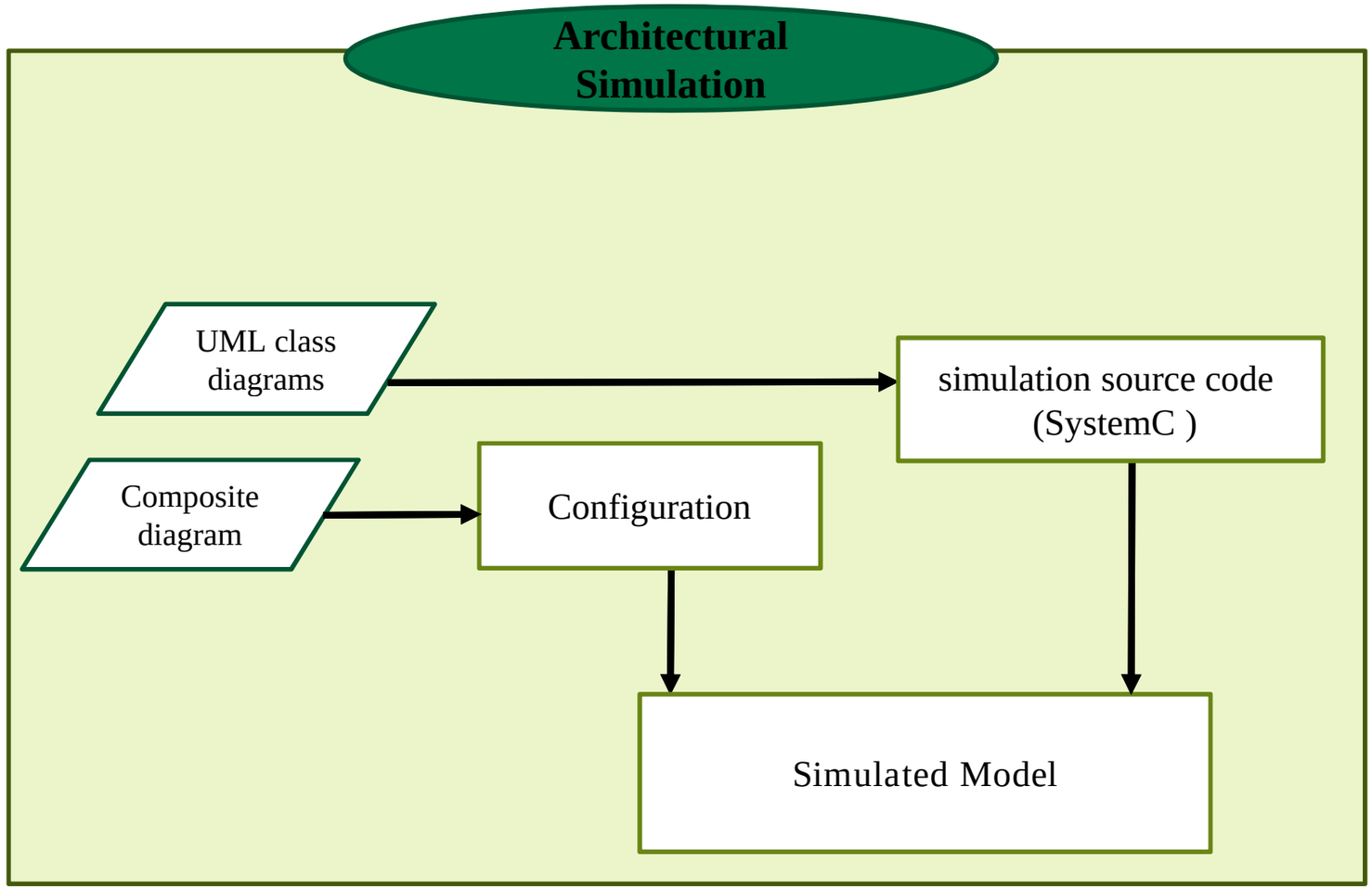


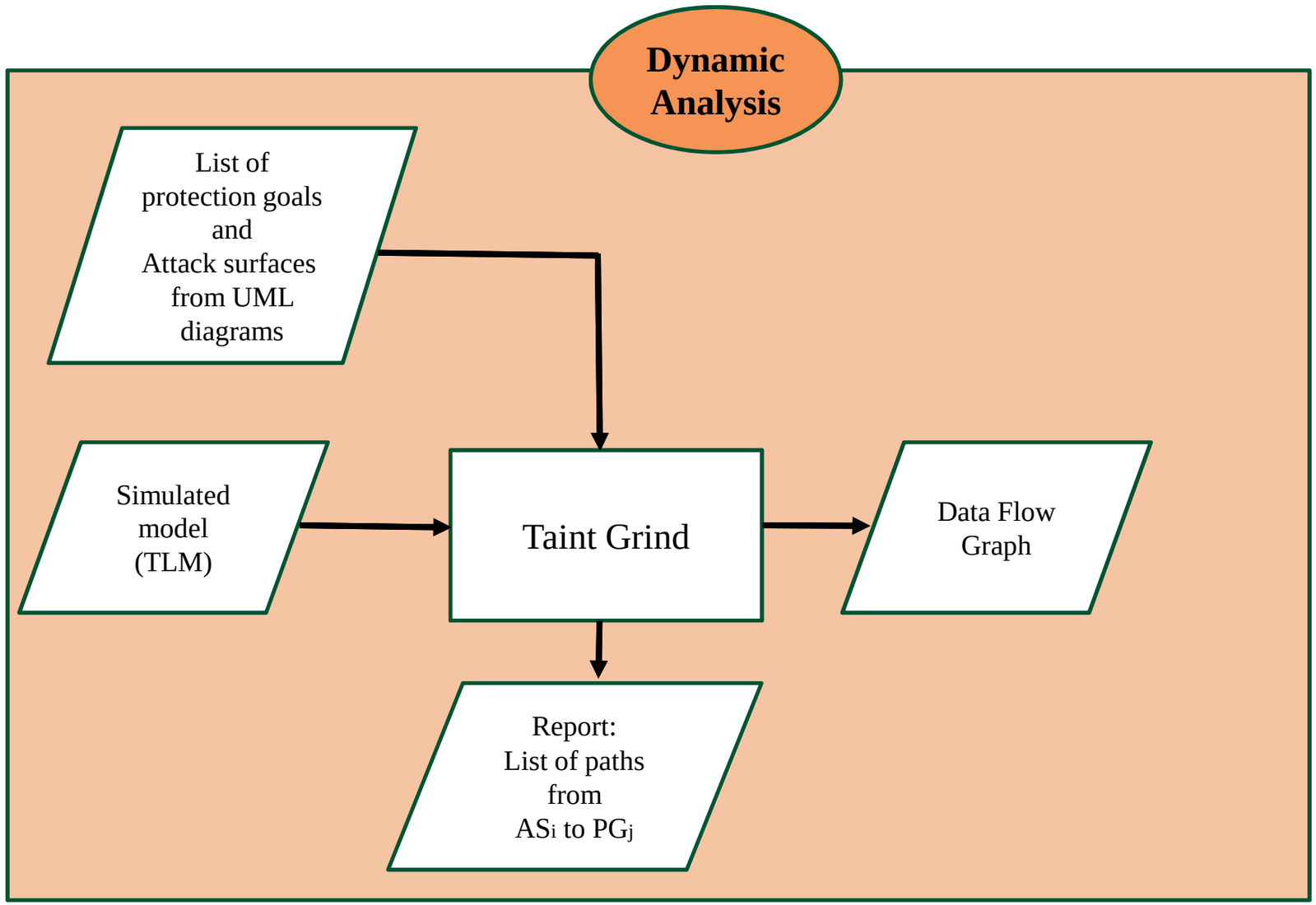


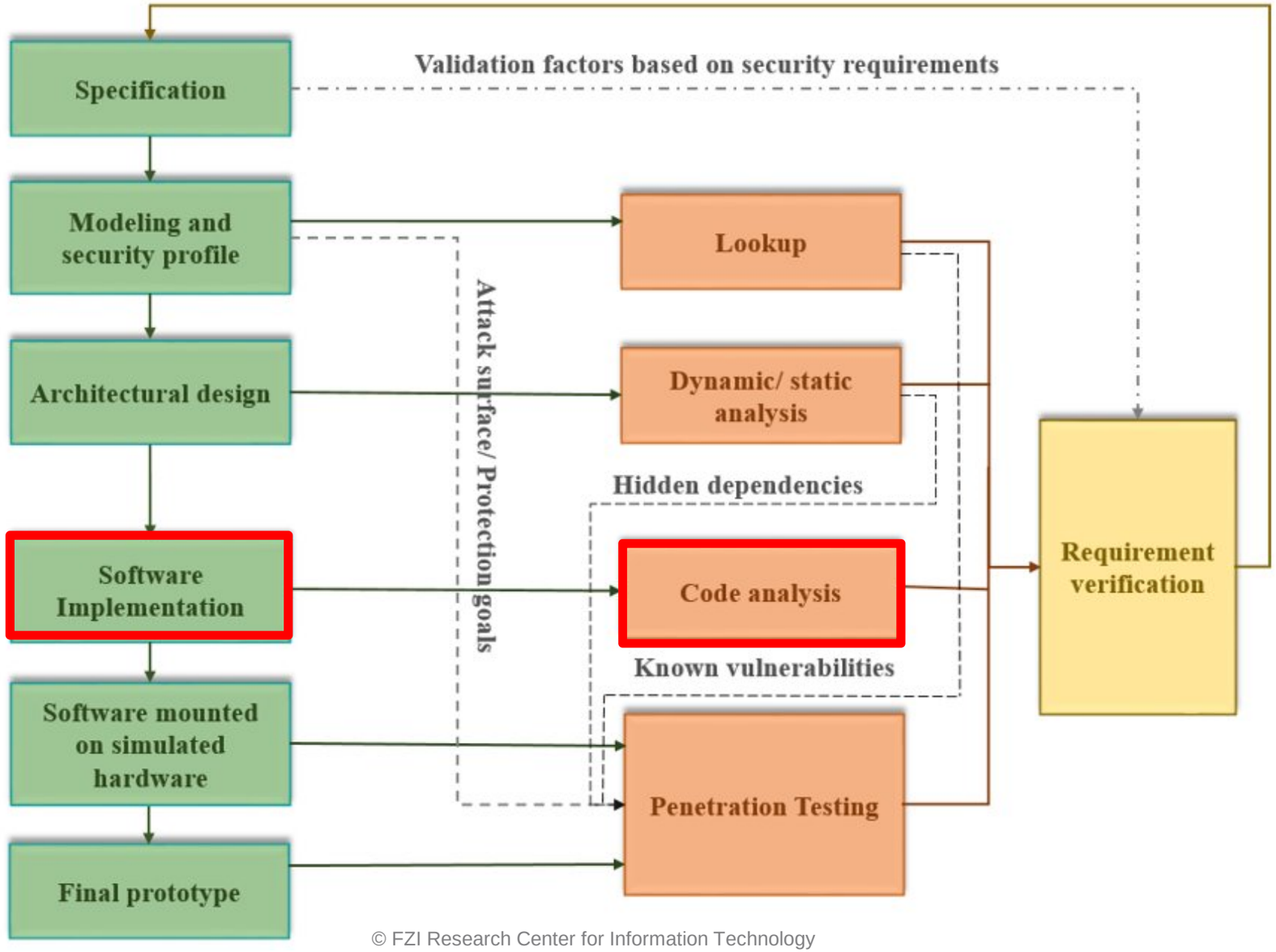








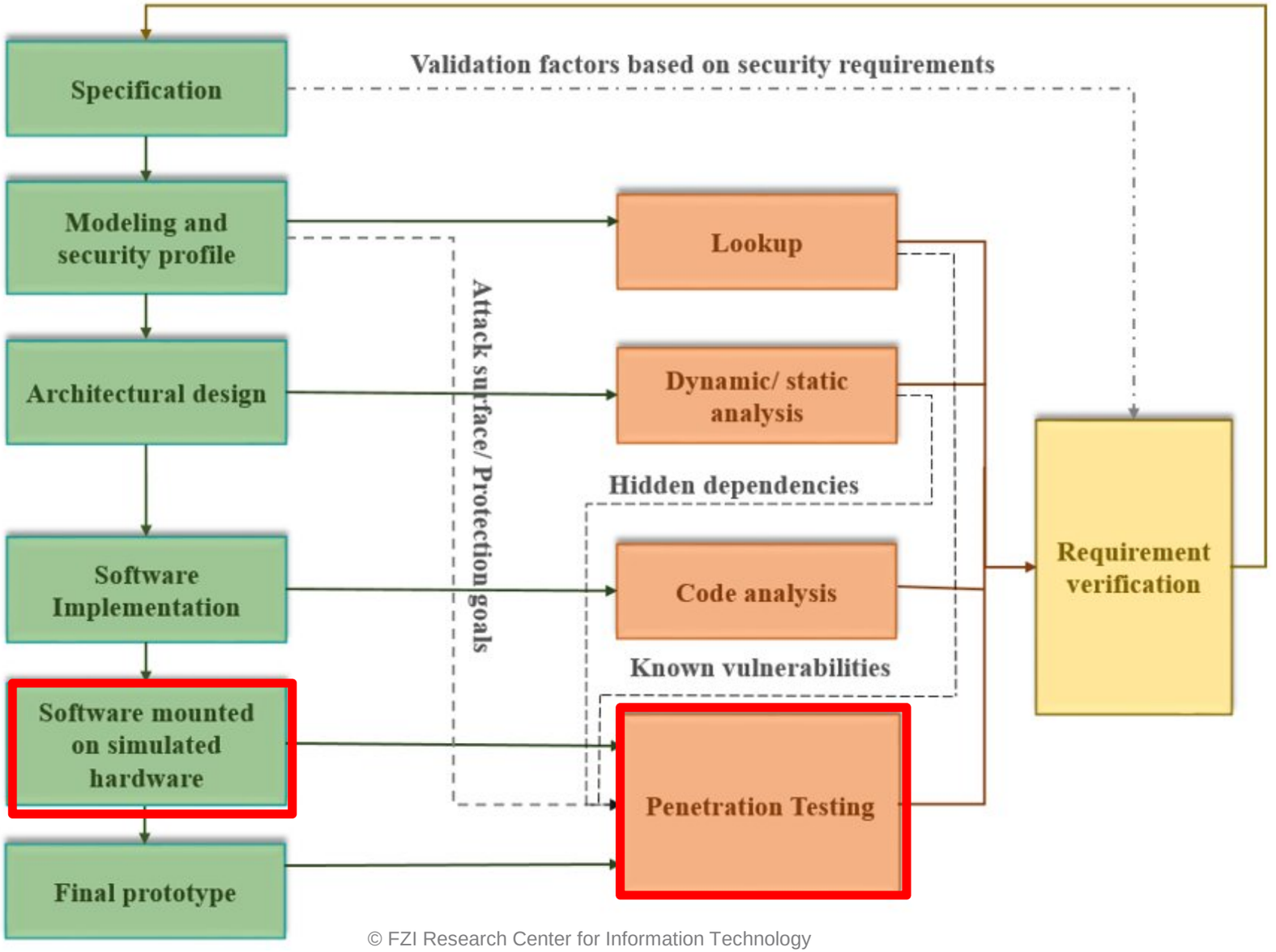


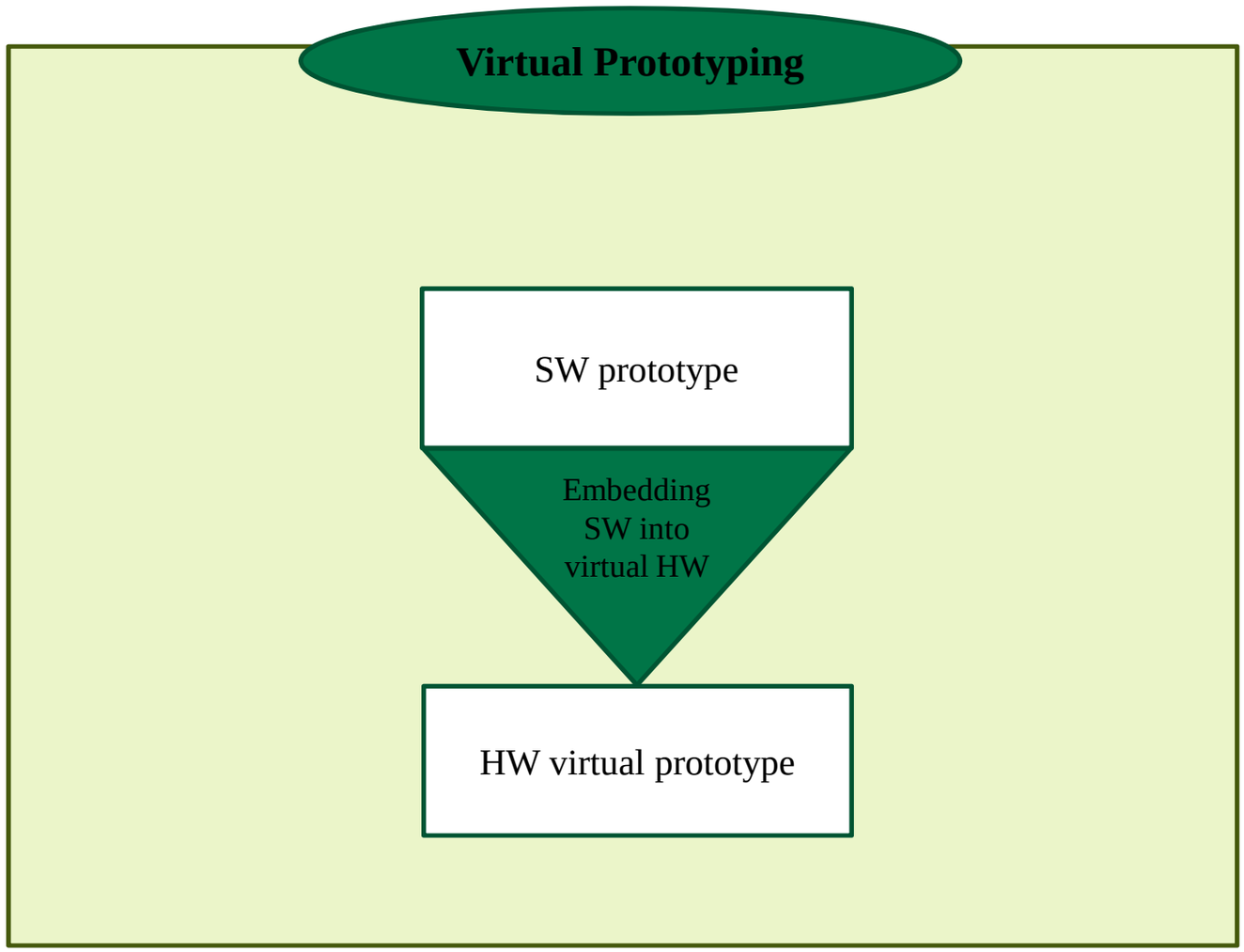


Code analysis

- Code vulnerabilities:
  - Security-injection flaws
  - Security-configuration flaws

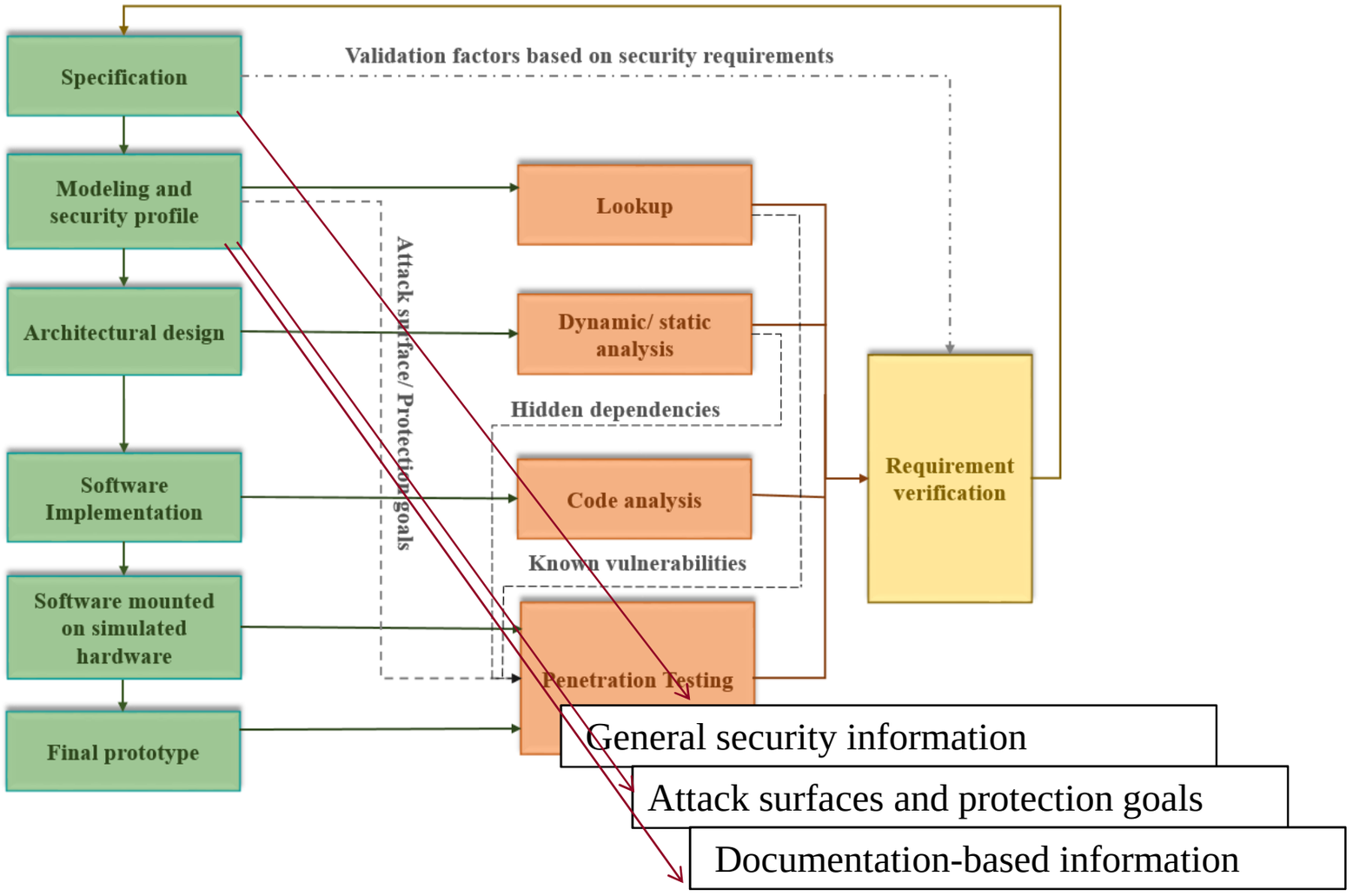
- Code analysis tools:
  - SonarQube
  - Deepsource
  - Veracode





Penetration Testing

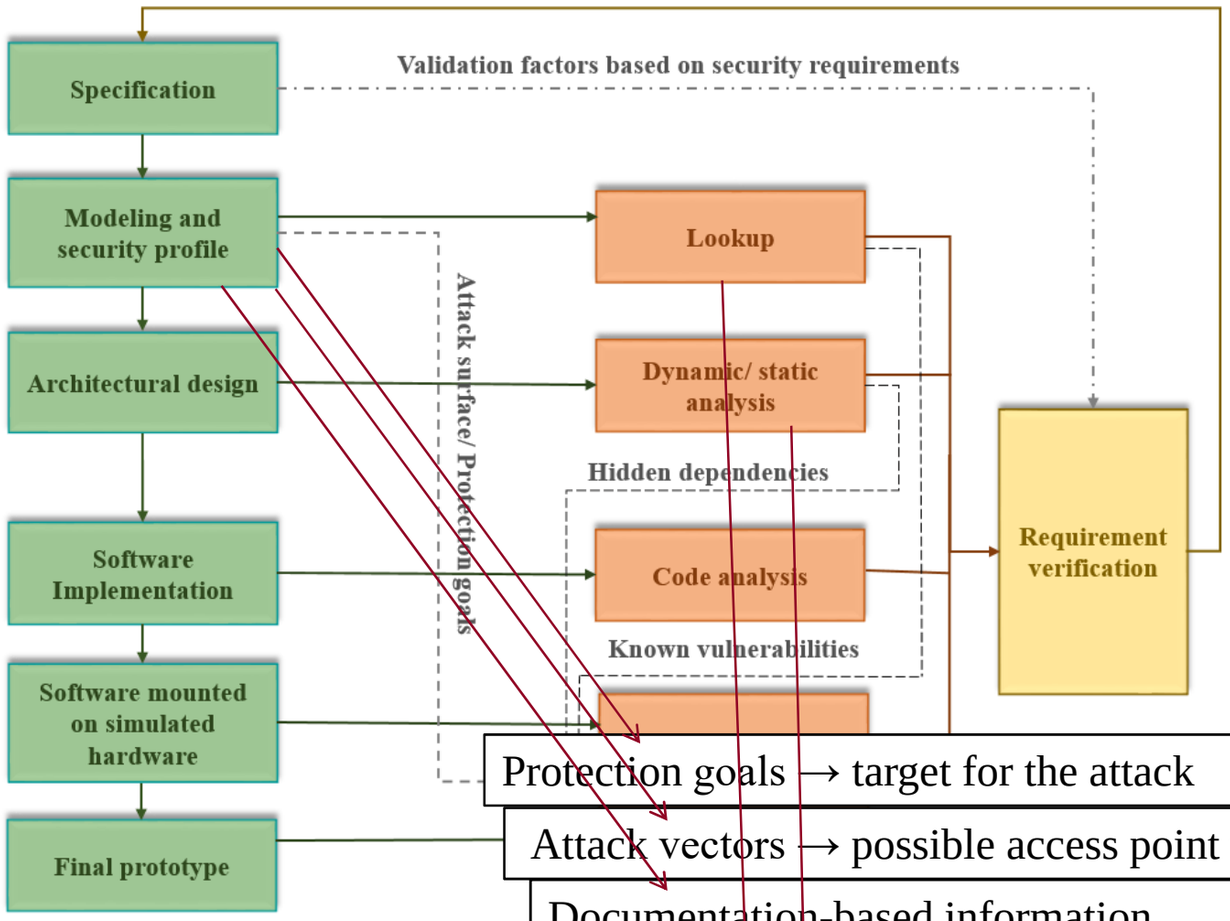
Recognition



Penetration Testing

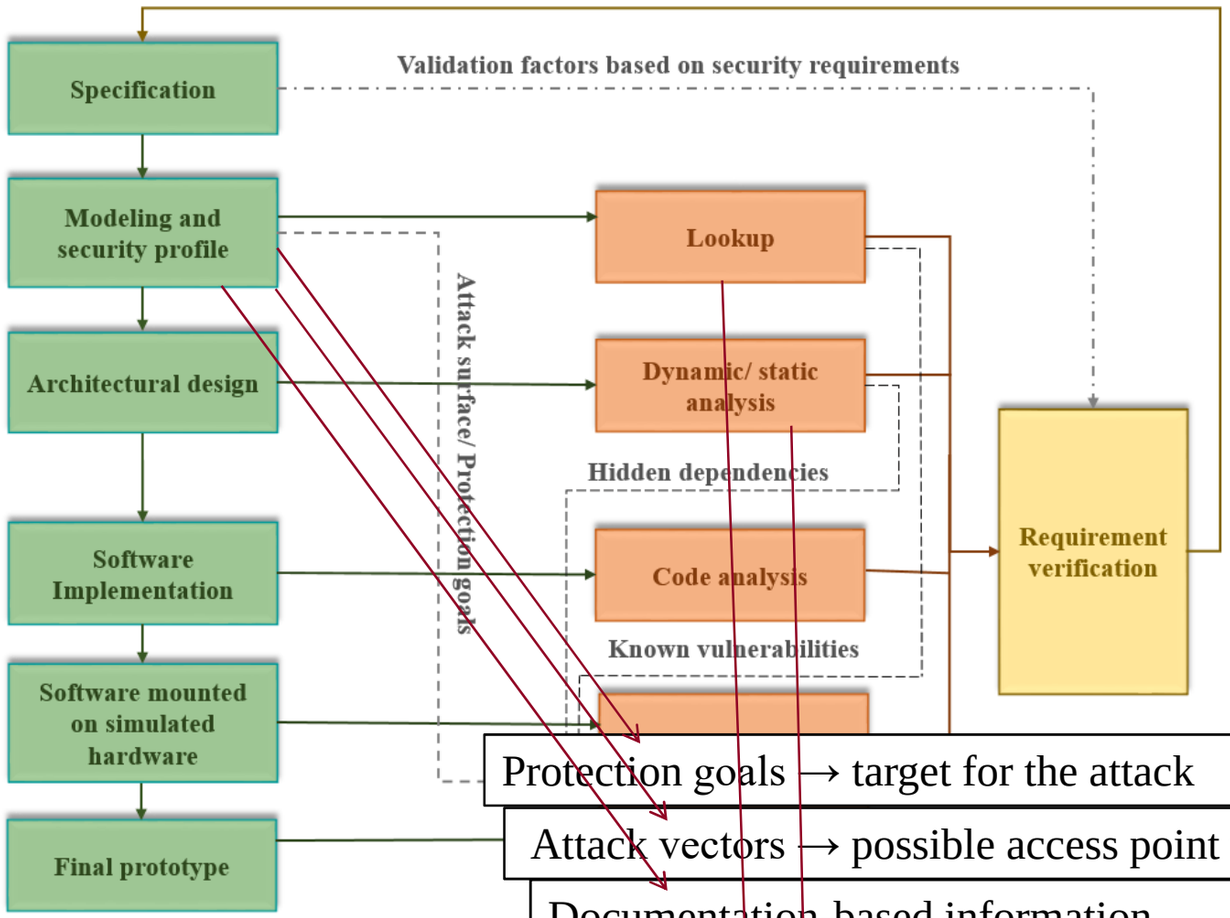
Recognition

Scanning



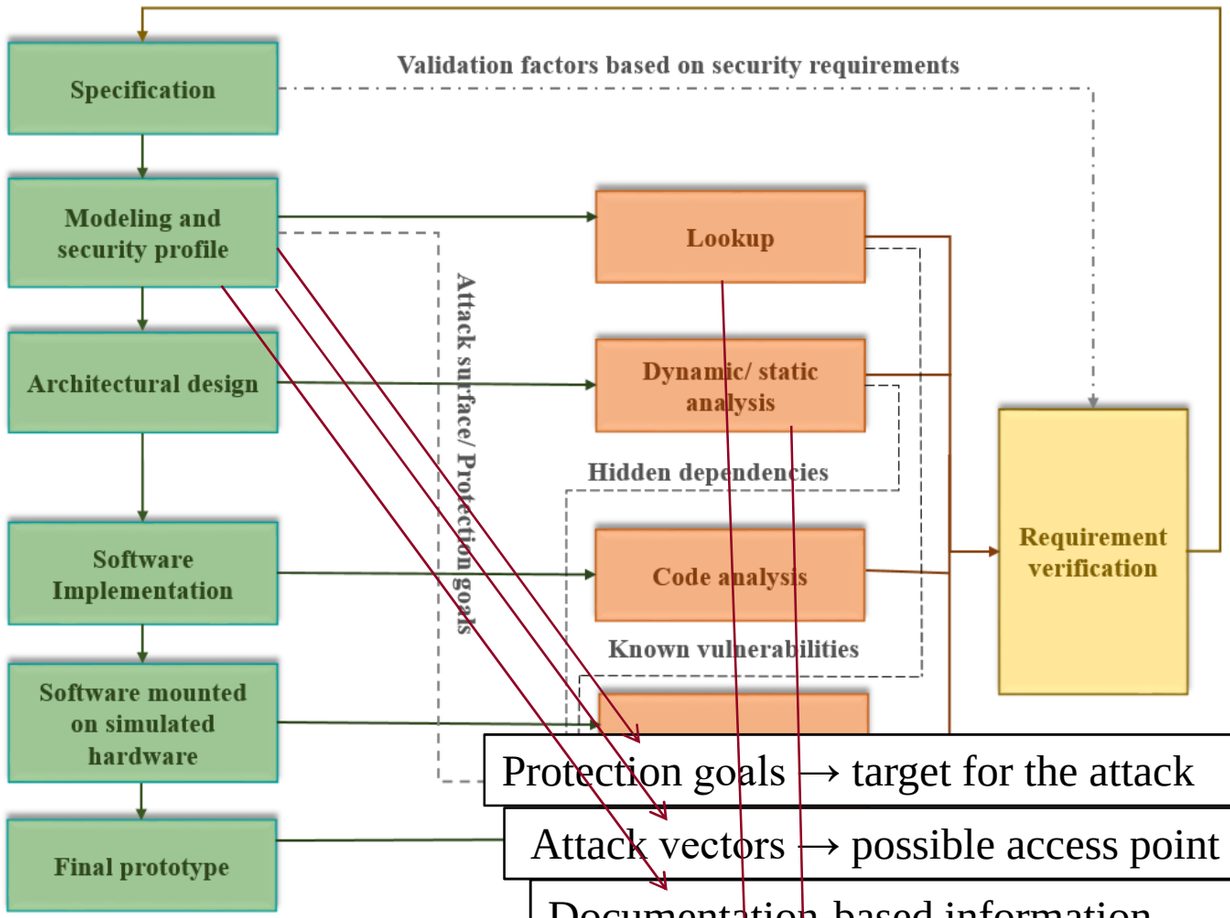
Penetration Testing

- Recognition
- Scanning
- Gaining Access



Penetration Testing

- Recognition
- Scanning
- Gaining Access
- Exploitation



Protection goals → target for the attack

Attack vectors → possible access point to the system

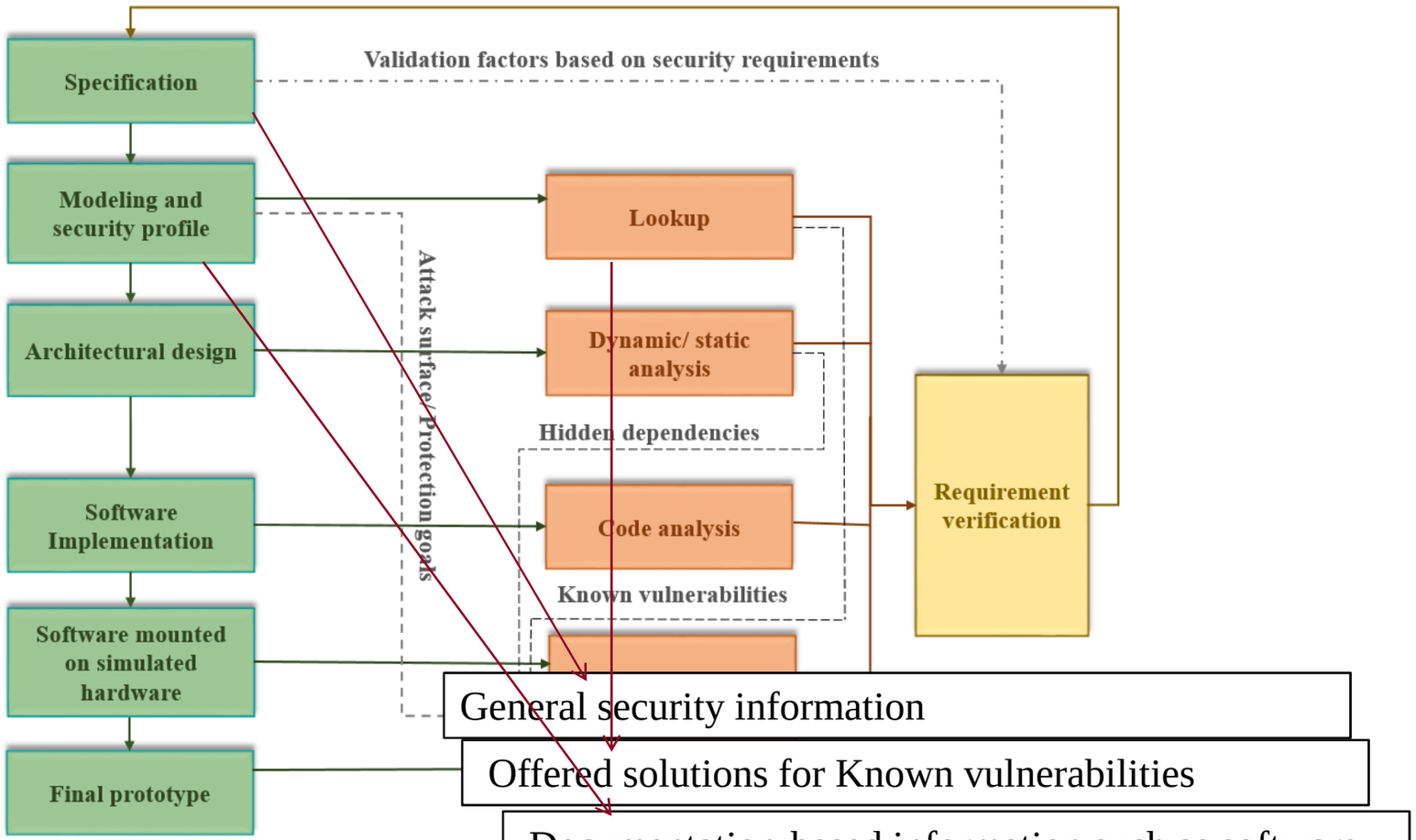
Documentation-based information

Known vulnerabilities → exploit the system

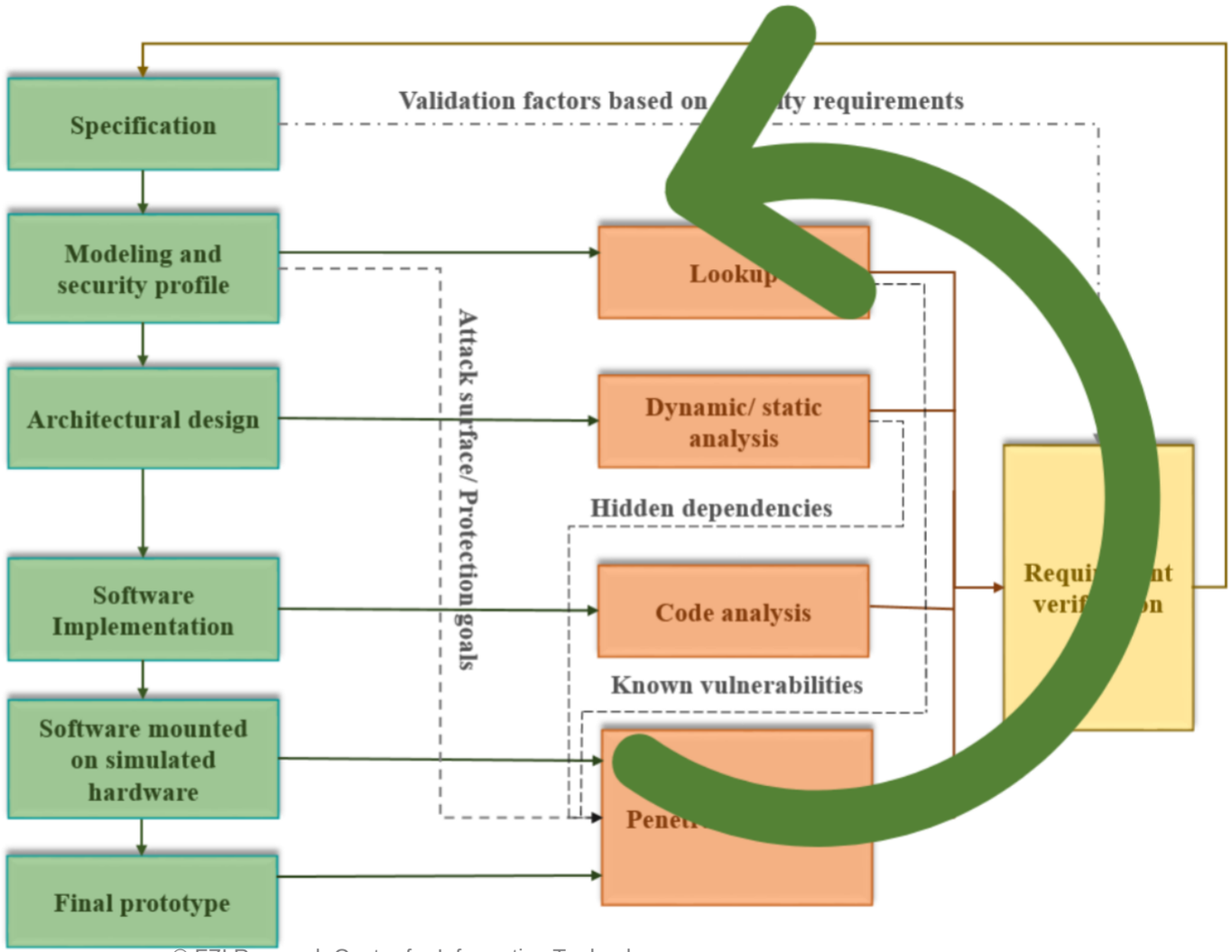
Hidden dependencies → exploitation

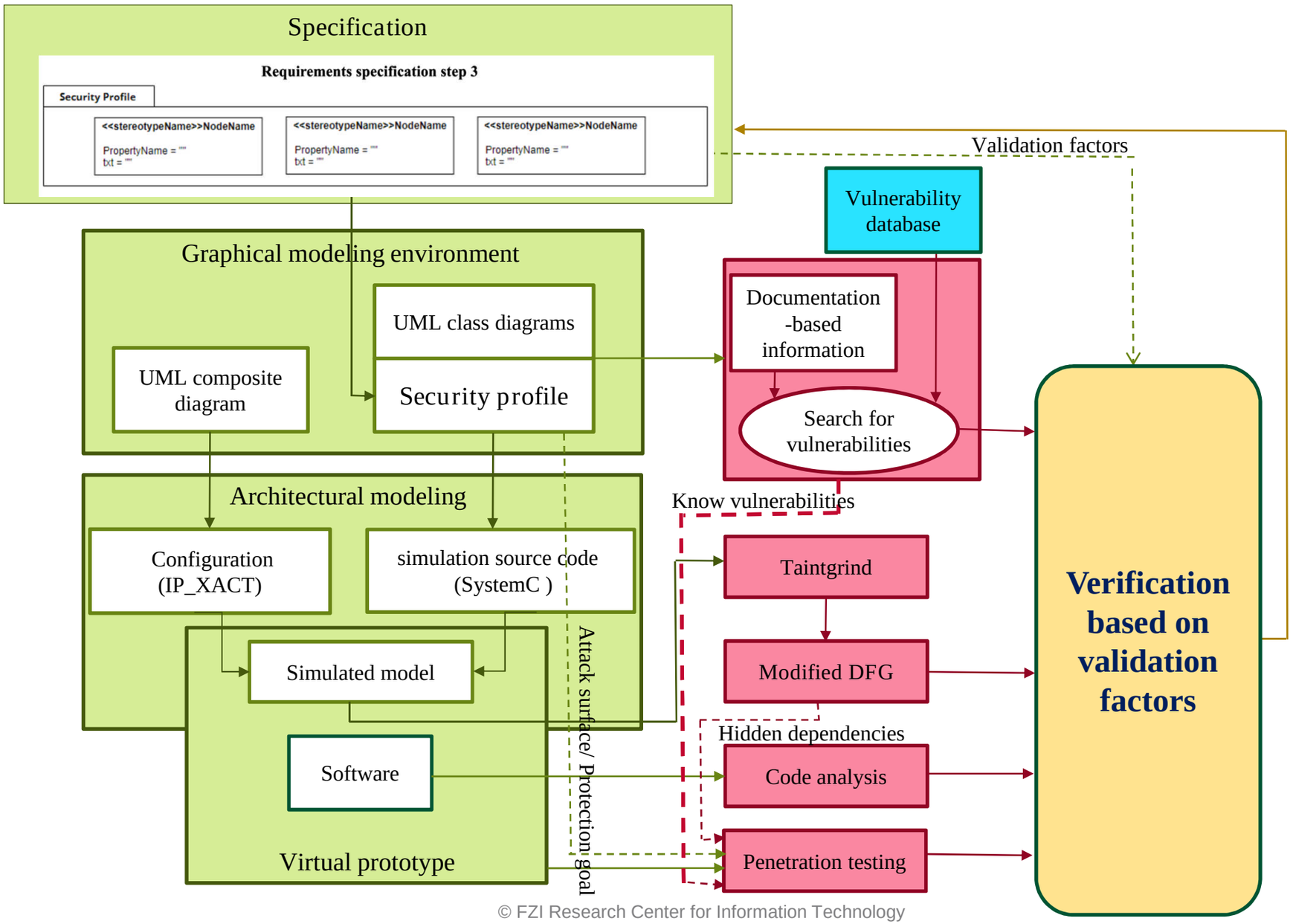
Penetration Testing

- Recognition
- Scanning
- Gaining Access
- Exploitation
- Reporting



Closed loop





Advantages of presented methodology

- Cost efficient early penetration testing
- Design decision verification
- Model verification
- Evaluation of security mechanisms performance
- Prevention strategy instead of reaction strategy
- Compliance with UMLsec

References

- [1] L. A. Bygrave, “Security by design: Aspirations and realities in a regulatory context,” Oslo Law Review, no. 3, pp. 126–177, 2022.
- [2] M. A. Rather and M. V. Bhatnagar, “A comparative study of software development life cycle models,” International Journal of Application or Innovation in Engineering & Management (IJAIEM), vol. 4, no. 10, pp. 23–29, 2015.
- [3] K. Pohl, M. Broy, H. Daembkes, and H. Honninger, “Advanced model- ¨ based engineering of embedded systems,” in Advanced Model-Based Engineering of Embedded Systems. Springer, 2016, pp. 3–9.
- [4] N. M. Mohammed, M. Niazi, M. Alshayeb, and S. Mahmood, “Exploring software security approaches in software development lifecycle: A systematic mapping study,” Computer Standards & Interfaces, vol. 50, pp. 107–115, 2017.
- [5] M. U. A. Khan and M. Zulkernine, “Quantifying security in secure software development phases,” in 2008 32nd Annual IEEE International Computer Software and Applications Conference. IEEE, 2008, pp. 955– 960.
- [6] S. Tverdyshev, “Security by design: Introduction to mils.” in MILS, 2017.
- [7] F. Koster, M. Klaas, H. Q. Nguyen, W. Brenner, M. Br ¨ andle, and ¨ S. Obermeier, “Collaborative security assessments in embedded systems development,” 2009.

References

- [8] J. Geismann, C. Gerking, and E. Bodden, “Towards ensuring security by design in cyber-physical systems engineering processes,” in Proceedings of the 2018 international conference on software and system process, 2018, pp. 123–127.
- [9] A. Ferrante, J. Milosevic, and M. Janjusevi ˇ c, “A security-enhanced ´ design methodology for embedded systems,” in 2013 International Conference on Security and Cryptography (SECRYPT). IEEE, 2013, pp. 1–12.
- [10] E. Mougoue, “What is the secure software development life cycle (sdlc)?— synopsys,” 2016.
- [11] M. E. Whitman and H. J. Mattord, “Principles of information security, course technology,” Google Scholar Google Scholar Digital Library Digital Library, 2012.
- [12] Y. Mahmoodi, S. Reiter, A. Viehl, O. Bringmann, and W. Rosenstiel, “Model-guided security analysis of interconnected embedded systems.” in MODELSWARD, 2018, pp. 602–609.
- [13] Ibm internet security systems. x-force. [Online]. Available: https://www.ibm.com/x-force [14] Open source vulnerability database.
- [15] Cve list. [Online]. Available: http://cve.mitre.org/
- [16] M. G. Kang, S. McCamant, P. Poosankam, and D. Song, “Dta++: dynamic taint analysis with targeted control-flow propagation.” in NDSS, 2011.
- [17] I. S. Association et al., “Standard systemc language reference manual,” IEEE Std, pp. 1666–2011, 2011.

Thank you for your attention