# Identifying and Measuring Modularity Violations in Cyber-Physical Systems

## Sponsor: DASD(SE)

**By**
**Lu Xiao and Michael Pennock**
**9th Annual SERC Sponsor Research Review**
**November 8, 2017**
**FHI 360 CONFERENCE CENTER**
**1825 Connecticut Avenue NW, 8th Floor**
**Washington, DC 20009**

**www.sercuarc.org**

# Talk Agenda



- *What is Modularity Violation (MV) in CPS*
- *Challenges in Identifying MV in CPS*
- *Pilot Study Approach and Results*
- *Future Research Tasks*

# Talk Agenda



- ✓ ***What is Modularity Violation (MV) in CPS***

- ❑ *Challenges in Identifying MV in CPS*

- ❑ *Pilot Study Approach and Results*

- ❑ *Future Research Tasks*

# Cyber-physical Systems (CPS)



- The term cyber-physical system is first coined in 2006. A cyber-physical system is an integration of computation with physical processes.

- *Physical* and *software* components are deeply *intertwined and interacting* with each other under changing context.

- Cyber-physical systems have gained wide-spread application in diverse areas including: civil infrastructure, energy, healthcare, transportation, automotive, smart appliances, and others

# Modularization in CPS



CPS = Physical + Software modules.

Modularization = Interdependence within

+ Independence across modules

- ✓ Improve Interoperability
- ✓ Facilitate system evolution and technology insertion
- ✓ Prevent vender lock-in and foster competition

# What is a module?

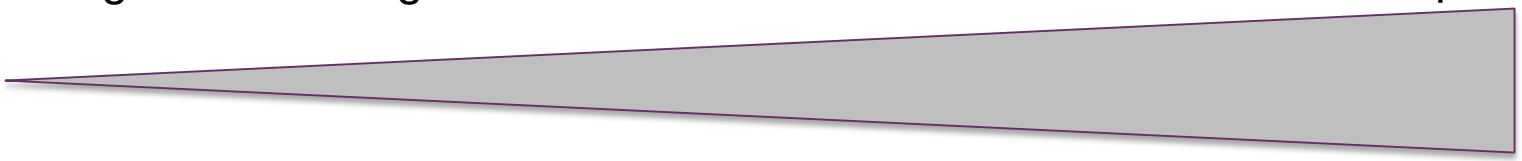


- The definition of a module will depend on the nature of system and the goals of different stakeholders.

Detail-oriented developers may view modules as hands-on and manageable working elements

Coupling and Cohesion

A project owner views modules as cohesive functional components to deliver value and competitiveness.

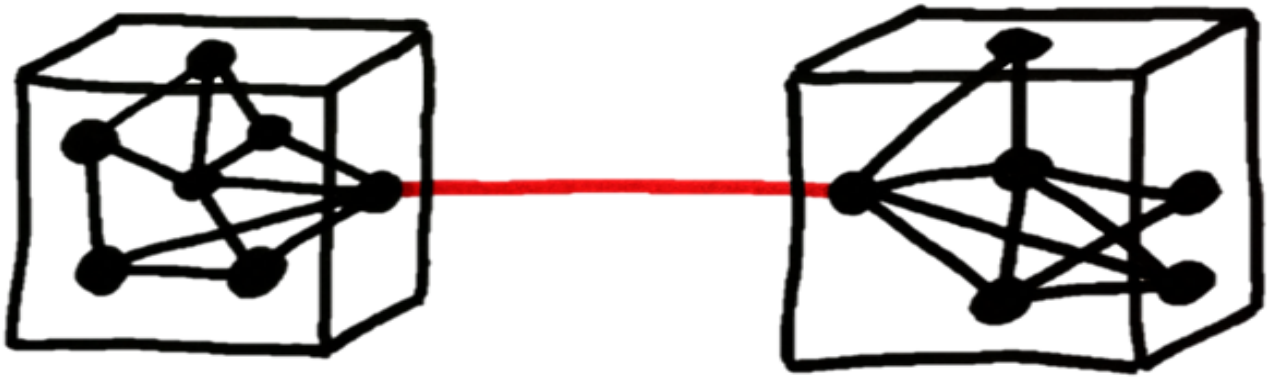


- There is a need to be able to handle modules of different granularity in the analysis.

# Modularity *Violations*



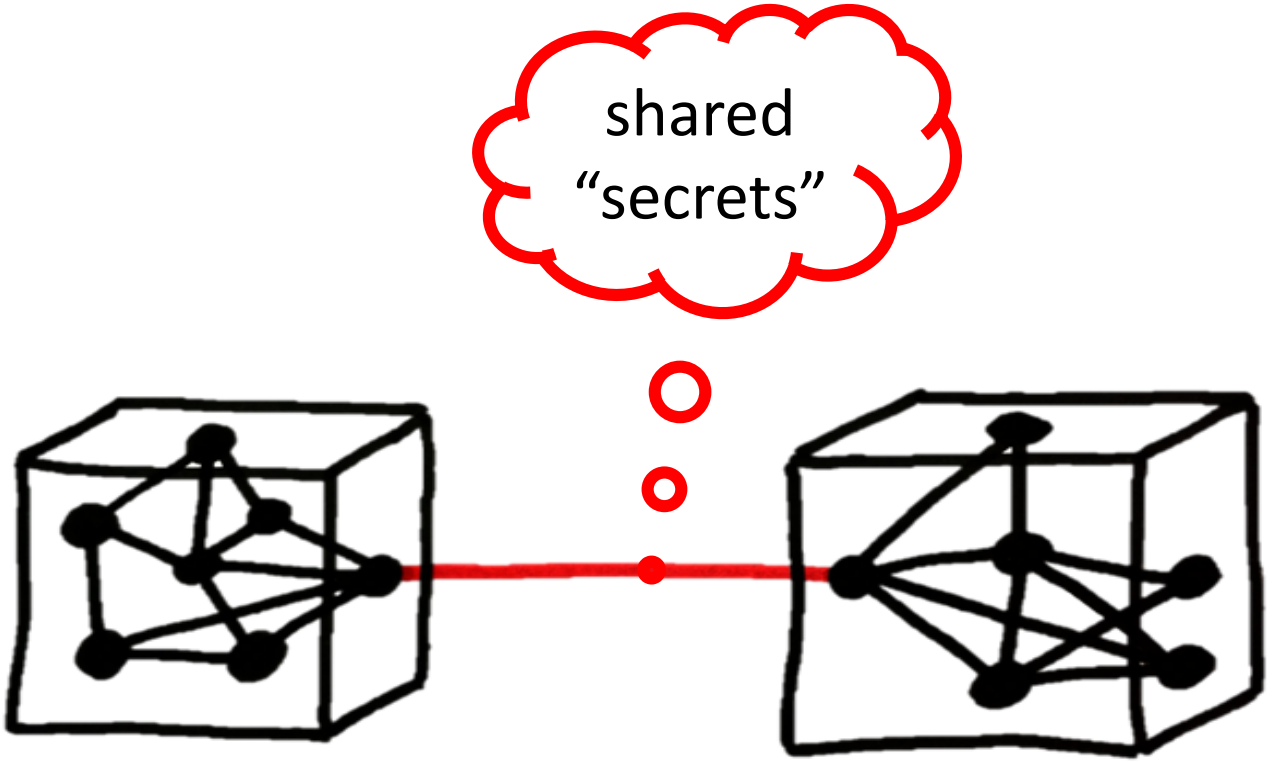
- Two supposedly independent modules actually coupled in the evolution of the system.
  - For example, update to one module requires corresponding update to another module.



# Modularity *Violations*



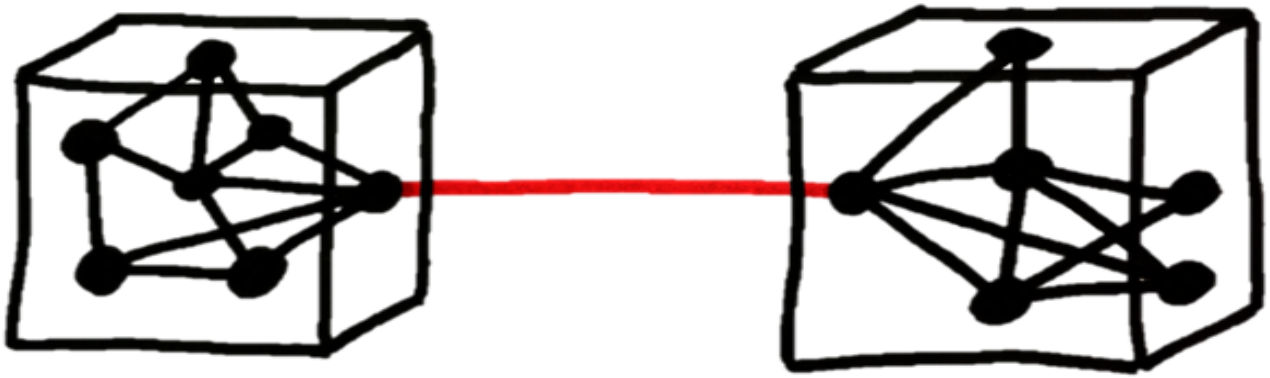
- Two supposedly independent modules actually coupled in the evolution of the system.
  - For example, update to one module requires corresponding update to another module.



# Modularity *Violations in CPS*



- Two supposedly independent modules actually coupled in the evolution of the system.
  - For example, update to one module requires corresponding update to another module.
    - Software vs. Software: SS-MV
    - Hardware vs. Software: HS-MV
    - Hardware vs. Hardware: HH-MV



# Talk Agenda

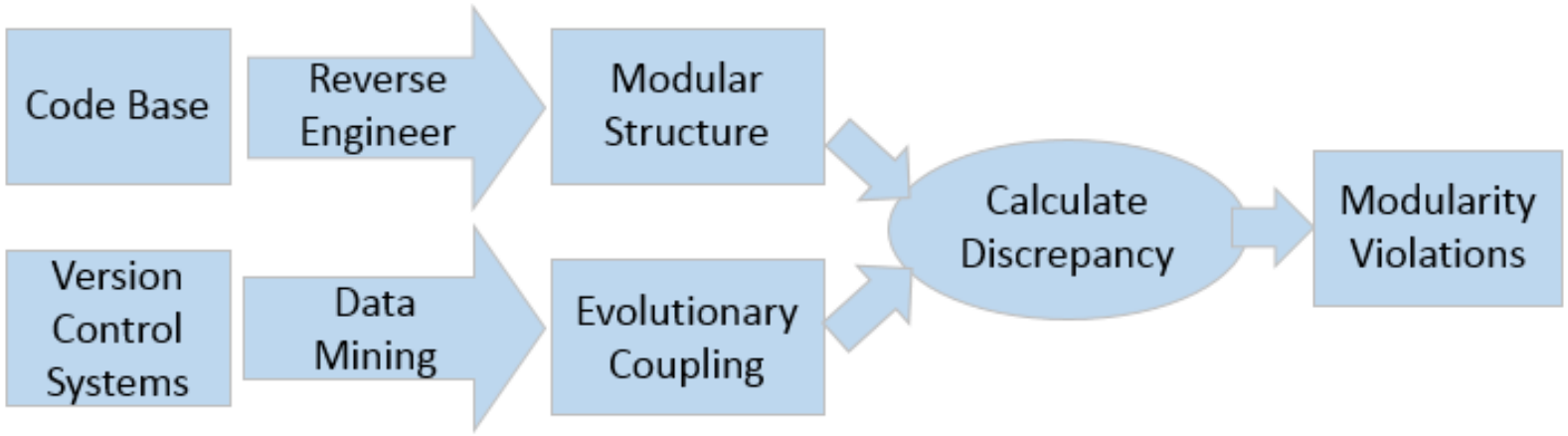


- ❑ *What is Modularity Violation (MV) in CPS*
- ✓ ***Challenges in Identifying MV in CPS***
- ❑ *Pilot Study Approach and Results*
- ❑ *Future Research Tasks*

# Approach for Software Systems



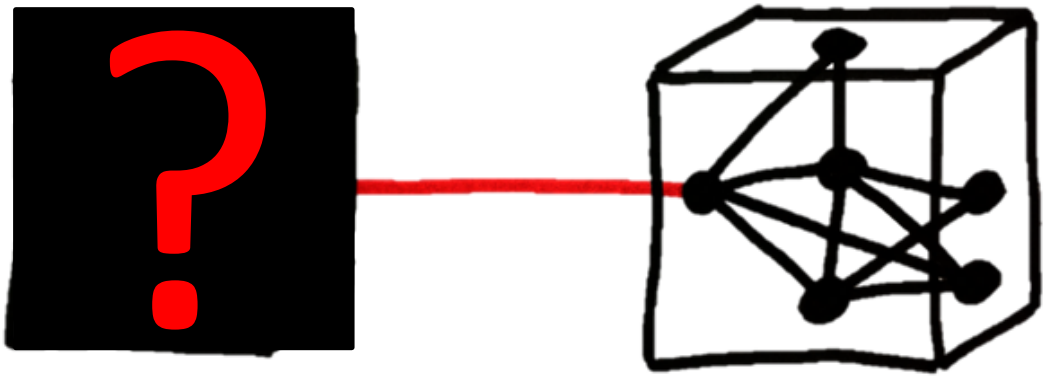
1. *Reverse-engineering to recover the modular structure*
2. *Data Mining to extract the co-change coupling*
3. *The discrepancy points to MV*



# Challenges for CPS



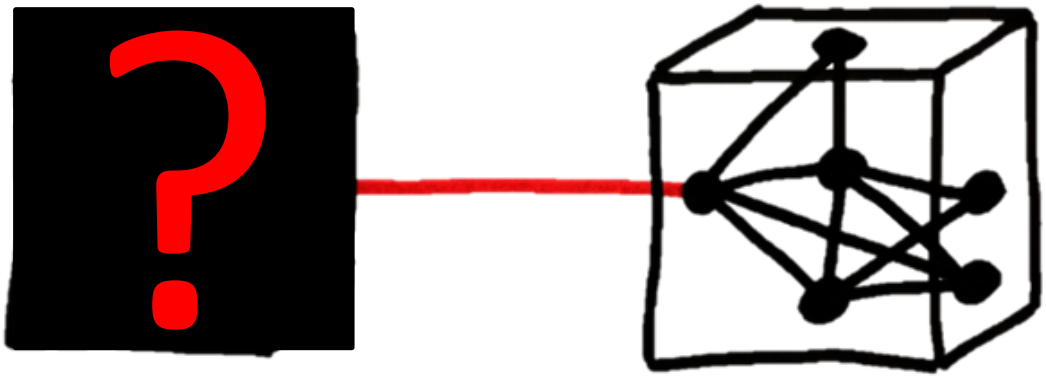
- It has been recognized as a major challenge to model software and hardware components in one “picture”.
- The maintenance data for the hardware side is like a “black box”.



# Challenges for CPS



- It has been recognized as a major challenge to model software and hardware components in one “picture”.
- The maintenance data for the hardware side is like a “black box”.
- **Is it feasible to investigate MV in CPS at all?**
- **How to identify MV involving hardware in CPS?**



# Talk Agenda



- ❑ *What is Modularity Violation (MV) in CPS*
- ❑ *Challenges in Identifying MV in CPS*
- ✓ ***Pilot Study Approach and Results***
- ❑ *Future Research Tasks*

# Pilot Study Subjects



OpenWrt: a Linux distribution for embedded devices that frees you from the application selection and configuration provided by the vendor.

MD PnP: the medical device “Plug-and-Play” interoperability program to improve patient care.

# Pilot Study Subjects



OpenWrt
Wireless Freedom

- 1052 files
- 6061 methods
- 163,114 LOC
- 137 software developers
- 38,099 revisions (2004~present)

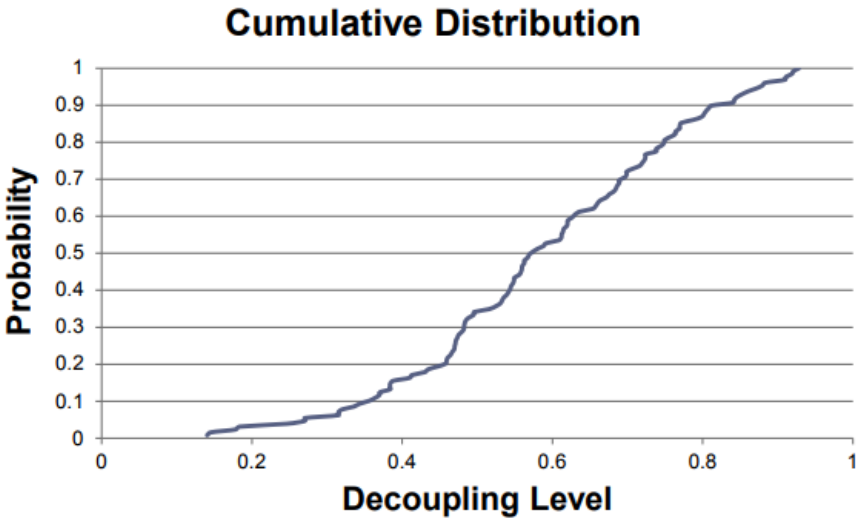
MD PnP™
Getting Connected for Patient Safety™

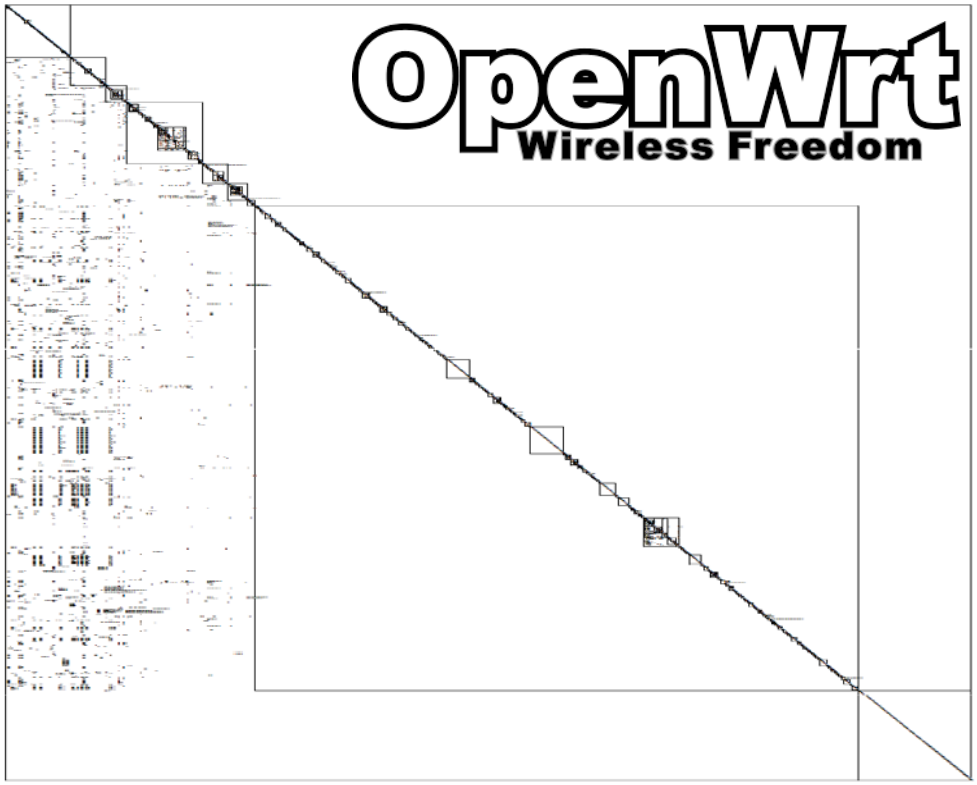
- 866 files
- 7872 methods
- 73,616 LOC
- 12 software developers
- 1605 revisions (2013~present)

# OpenWrt is Well Modularized



- 1052 files
- 1.4% propagation cost
- 78% decoupling level
  - 108 open source, 21 industrial

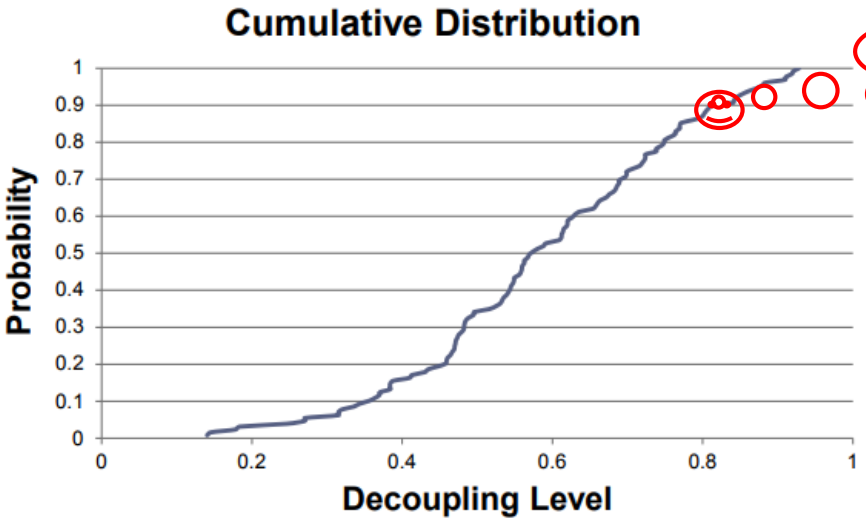


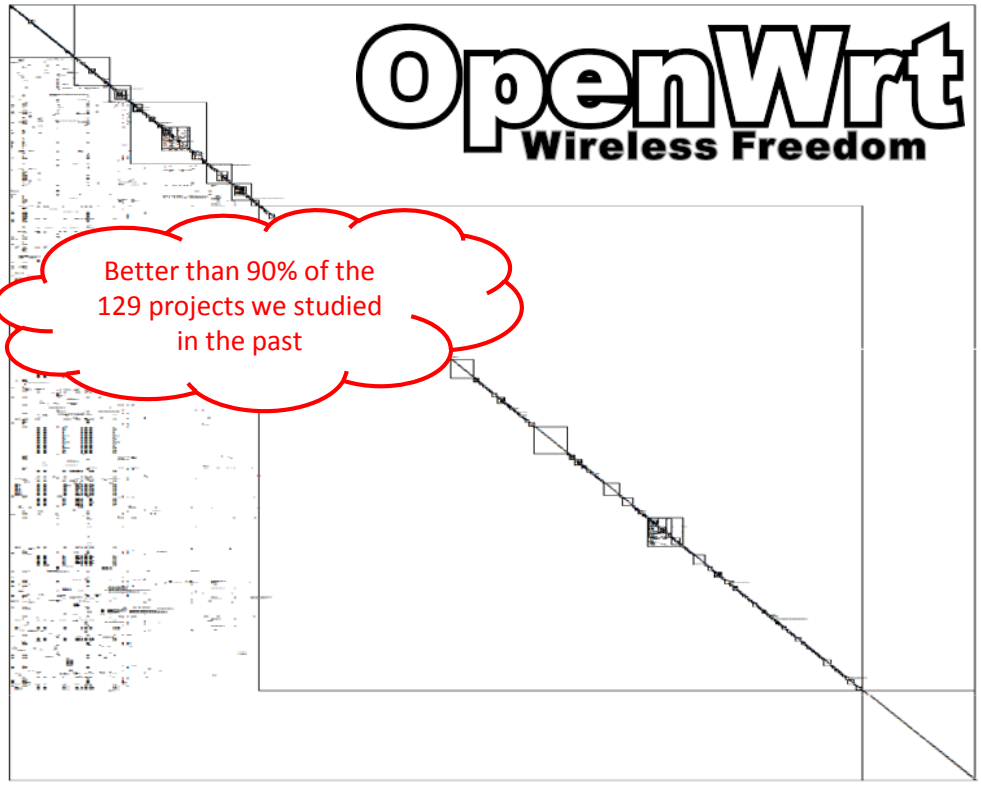



# OpenWrt is Well Modularized



- 1052 files
- 1.4% propagation cost
- 78% decoupling level
  - 108 open source, 21 industrial

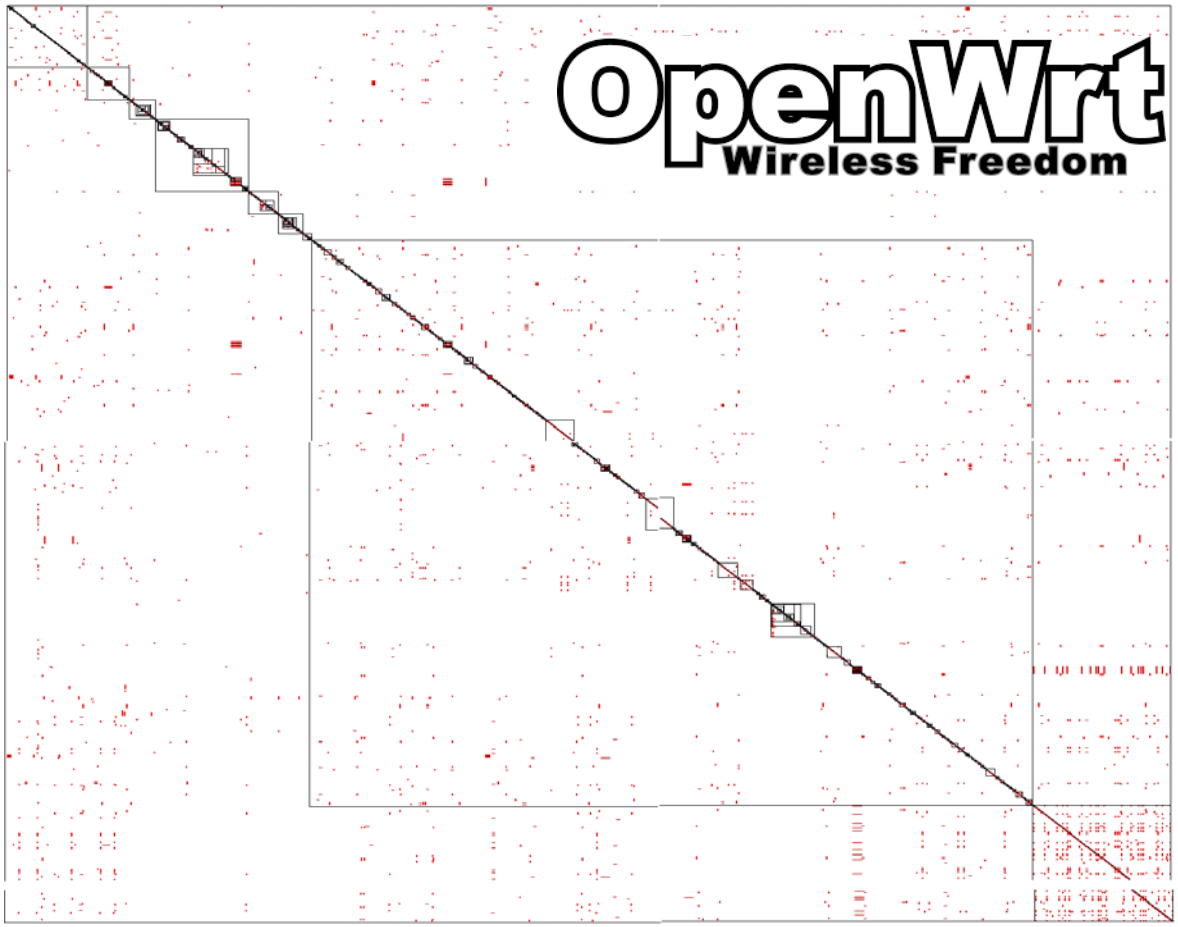


# Data Mining: Co-Change Coupling



- 38,099 revisions (2004~present)
- 4228 co-changed pairs
- 1 to 6 change frequency



# Software Modularity Violations



- co-change threshold 4
- 23 files involved in modularity violations

OpenWrt Wireless Freedom

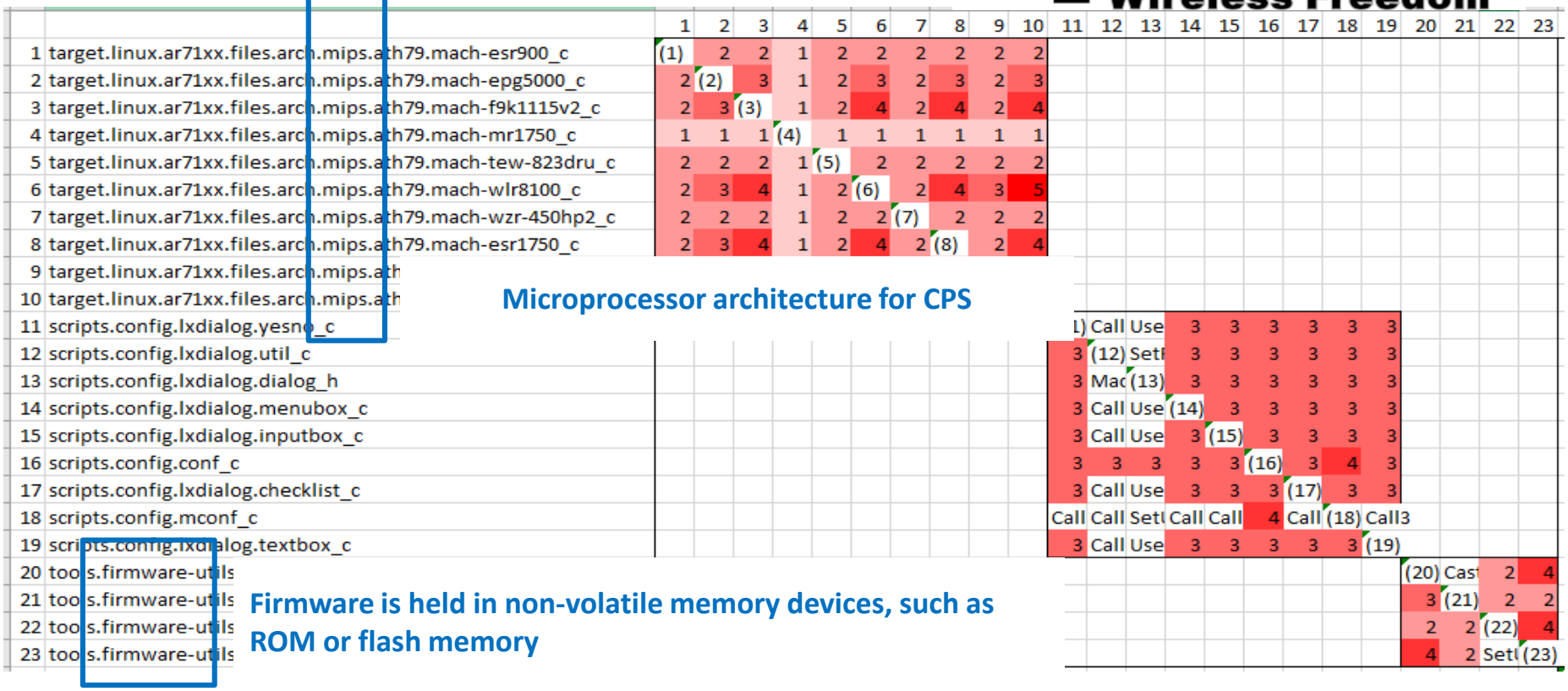
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | target.linux.ar71xx.files.arch.mips.ath79.mach-esr900_c | (1) | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | | | | | | | | | | | | | |
| 2 | target.linux.ar71xx.files.arch.mips.ath79.mach-epg5000_c | 2 | (2) | 3 | 1 | 2 | 3 | 2 | 3 | 2 | 3 | | | | | | | | | | | | | |
| 3 | target.linux.ar71xx.files.arch.mips.ath79.mach-f9k1115v2_c | 2 | 3 | (3) | 1 | 2 | 4 | 2 | 4 | 2 | 4 | | | | | | | | | | | | | |
| 4 | target.linux.ar71xx.files.arch.mips.ath79.mach-mr1750_c | 1 | 1 | 1 | (4) | 1 | 1 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | |
| 5 | target.linux.ar71xx.files.arch.mips.ath79.mach-tew-823dru_c | 2 | 2 | 2 | 1 | (5) | 2 | 2 | 2 | 2 | 2 | | | | | | | | | | | | | |
| 6 | target.linux.ar71xx.files.arch.mips.ath79.mach-wlr8100_c | 2 | 3 | 4 | 1 | 2 | (6) | 2 | 4 | 3 | 5 | | | | | | | | | | | | | |
| 7 | target.linux.ar71xx.files.arch.mips.ath79.mach-wzr-450hp2_c | 2 | 2 | 2 | 1 | 2 | 2 | (7) | 2 | 2 | 2 | | | | | | | | | | | | | |
| 8 | target.linux.ar71xx.files.arch.mips.ath79.mach-esr1750_c | 2 | 3 | 4 | 1 | 2 | 4 | 2 | (8) | 2 | 4 | | | | | | | | | | | | | |
| 9 | target.linux.ar71xx.files.arch.mips.ath79.mach-tl-wr1043nd-v2_c | 2 | 2 | 2 | 1 | 2 | 3 | 2 | 2 | (9) | 3 | | | | | | | | | | | | | |
| 10 | target.linux.ar71xx.files.arch.mips.ath79.mach-nbg6716_c | 2 | 3 | 4 | 1 | 2 | 5 | 2 | 4 | 3 | (10) | | | | | | | | | | | | | |
| 11 | scripts.config.lxdialog.yesno_c | | | | | | | | | | | (11) | Call | Use | 3 | 3 | 3 | 3 | 3 | 3 | | | | |
| 12 | scripts.config.lxdialog.util_c | | | | | | | | | | | 3 | (12) | Set | 3 | 3 | 3 | 3 | 3 | 3 | | | | |
| 13 | scripts.config.lxdialog.dialog_h | | | | | | | | | | | 3 | Mac | (13) | 3 | 3 | 3 | 3 | 3 | 3 | | | | |
| 14 | scripts.config.lxdialog.menubox_c | | | | | | | | | | | 3 | Call | Use | (14) | 3 | 3 | 3 | 3 | 3 | | | | |
| 15 | scripts.config.lxdialog.inputbox_c | | | | | | | | | | | 3 | Call | Use | 3 | (15) | 3 | 3 | 3 | 3 | | | | |
| 16 | scripts.config.conf_c | | | | | | | | | | | 3 | 3 | 3 | 3 | 3 | (16) | 3 | 4 | 3 | | | | |
| 17 | scripts.config.lxdialog.checklist_c | | | | | | | | | | | 3 | Call | Use | 3 | 3 | 3 | (17) | 3 | 3 | | | | |
| 18 | scripts.config.mconf_c | | | | | | | | | | | Call | Call | Set | Call | Call | 4 | Call | (18) | Call | 3 | | | |
| 19 | scripts.config.lxdialog.textbox_c | | | | | | | | | | | 3 | Call | Use | 3 | 3 | 3 | 3 | 3 | (19) | | | | |
| 20 | tools.firmware-utils.src.mkzynfw_c | | | | | | | | | | | | | | | | | | | | (20) | Cas | 2 | 4 |
| 21 | tools.firmware-utils.src.zynos_h | | | | | | | | | | | | | | | | | | | | 3 | (21) | 2 | 2 |
| 22 | tools.firmware-utils.src.csysimg_h | | | | | | | | | | | | | | | | | | | | 2 | 2 | (22) | 4 |
| 23 | tools.firmware-utils.src.mkcsysimg_c | | | | | | | | | | | | | | | | | | | | 4 | 2 | Set | (23) |

# Software Modularity Violations



- co-change threshold 4
- 23 files involved in modularity violations

OpenWrt Wireless Freedom

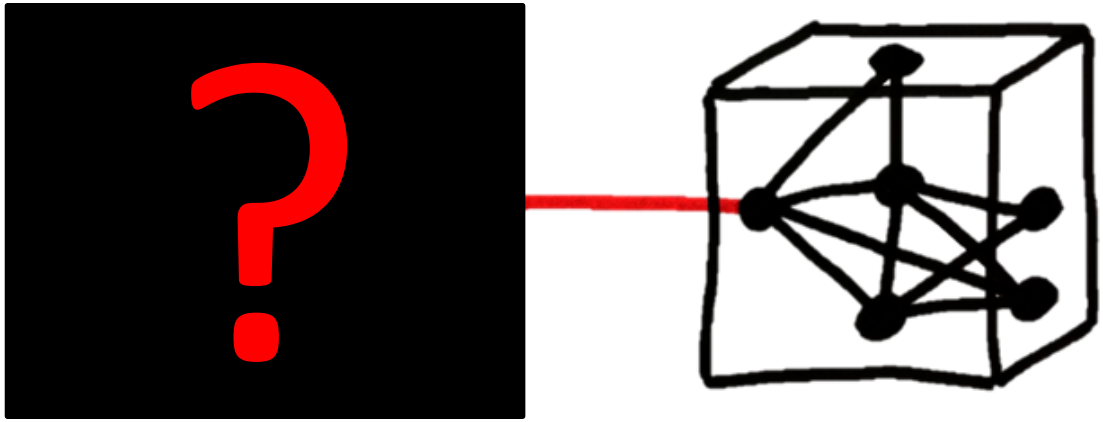



**Microprocessor architecture for CPS**

**Firmware is held in non-volatile memory devices, such as ROM or flash memory**

# Hardware-related MV

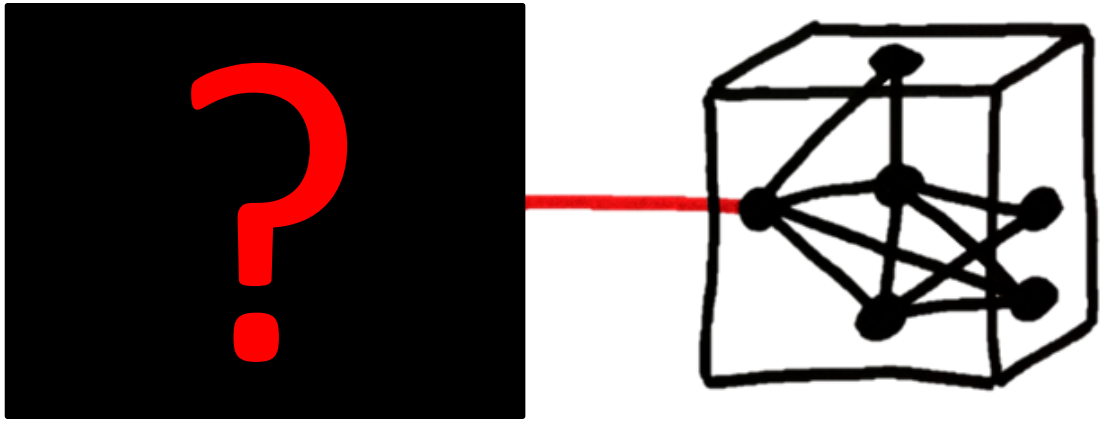





Whether and which software modules are likely to change due to hardware related concepts?

# Hardware-related MV





1. When developers make changes to a software module/entity, he/she says something about hardware components.

   "set *chip* type directly in ar8216_id_chip."

2. The naming of a software entity contains important hardware related key words, e.g. "mips", "firmware", etc..

   openwrt.tools.*firmware*-utils.src.mktplinkfw_c

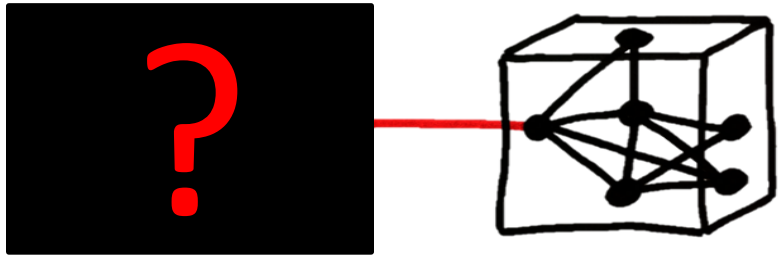
# Key-word based Heuristics



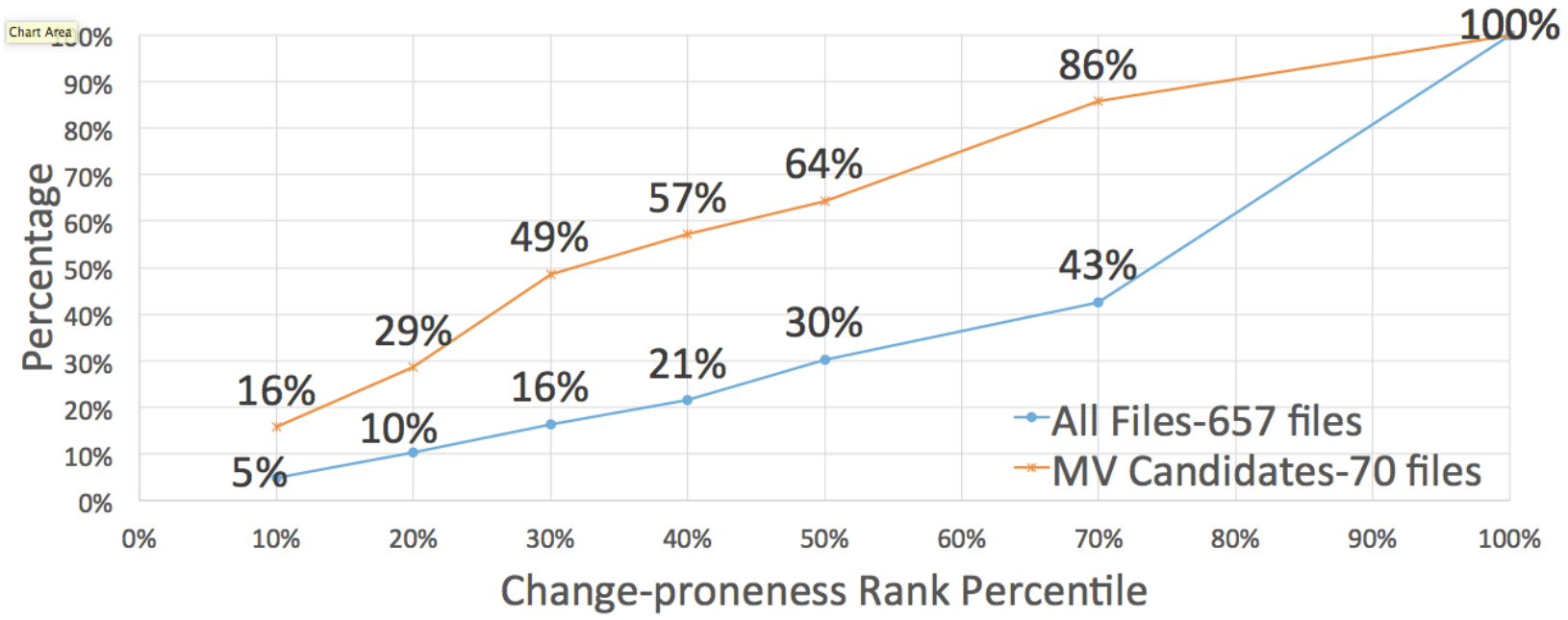
- 16 key words ***manually*** extracted from the commit messages and hardware devices supported by OpenWrt

"radio", "WiFi", "zigbee", "btle", "mips", "ramips", "mtd", "broadcom", "routerboot", "router", "firmware", "bluetooth", "energy", "power", "soc", "chip"

- 71 source files are extracted from the total 1052 files from OpenWrt, which are potential software-participants of hardware vs. software modularity violations.



# Hardware-related Concepts are Significant Change Contributors



# Talk Agenda



- *What is Modularity Violation (MV) in CPS*
- *Challenges in Identifying MV in CPS*
- *Pilot Study Approach and Results*
- ✓ ***Future Research Tasks***

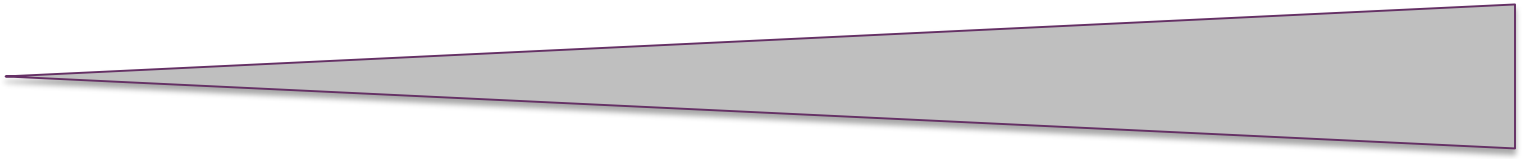
# Examine Module Criteria

- Modules for atomic working tasks.
- Modules based on coupling and/or cohesion
- Modules based on the system architecture.

Coupling and Cohesion

Detail-oriented developers may view modules as hands-on and manageable working elements

A project owner views modules as cohesive functional components to deliver value and competitiveness.



# Build Domain Concept "Leaner"



- In the incubator phase, we realized that the keyword heuristics developed for one project domain may not be applied to another project domain.
- We plan to build a "Domain Concept Learner", which leverages natural language processing techniques to learn domain languages from available sources.
- Using the resulting concept dictionary under different domain contexts and hardware environments, we can apply the approach to identify potential modularity violations to CPS projects in different domains.

# Decision Framework Demonstrator



- We plan to incorporate the methods developed in prior steps into a decision framework to increase their utility for decision makers.
- This decision framework will accommodate the detection and analysis of modularity violations at different modular granularities and in different problem domains.
- The ultimate goal is to provide decision making support to improve modularity where appropriate and achieve the associated benefits.