MITSUBISHI ELECTRIC
Changes for the Better

A Trusted Approach to Design a Network Monitor

Koichi Shimizu, Teruyoshi Yamaguchi, Tsunato Nakai, Takeshi Ueda, Nobuhiro Kobayashi and Benoît Boyer

MITSUBISHI ELECTRIC CORPORATION

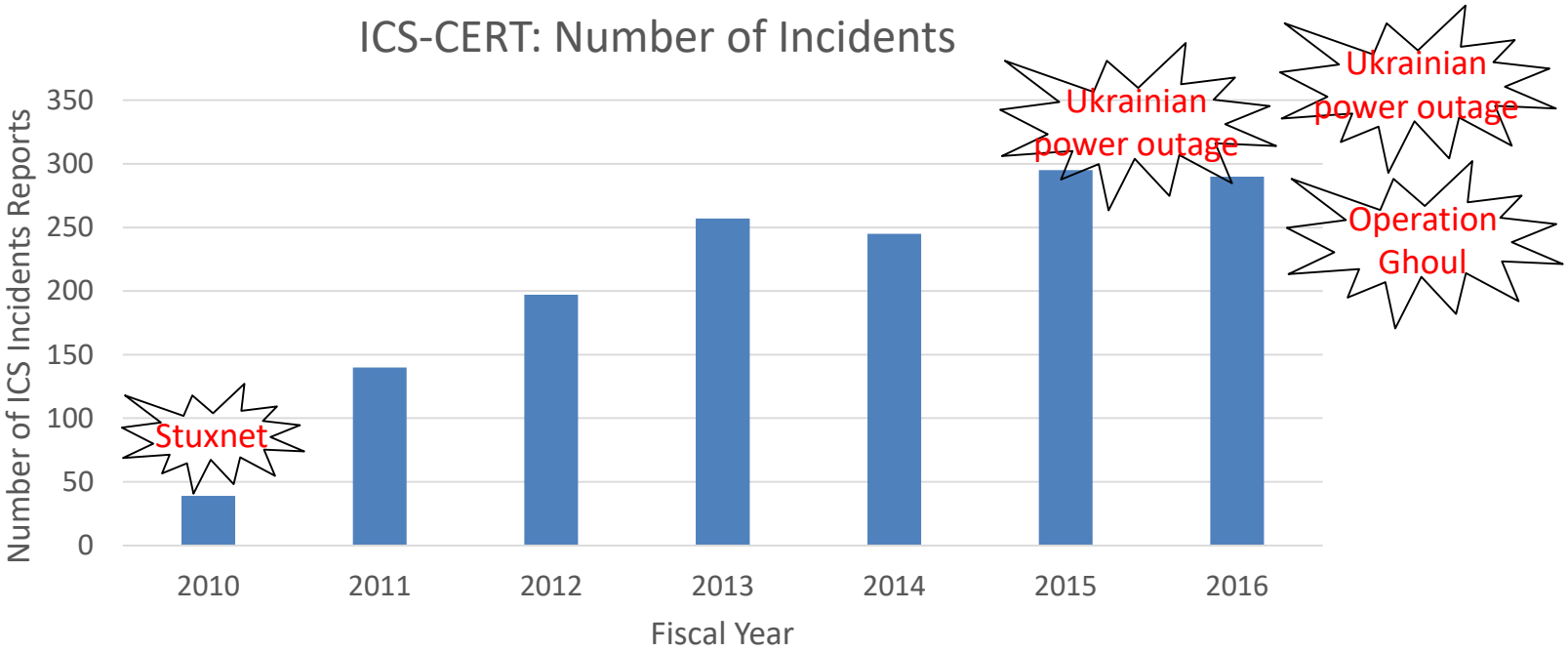
Outline

- Background
- Whitelisting Network Monitor
- Trusted Approach for Whitelisting
  - Model-based
  - Automation
  - Verification
- Conclusion

Background

Cyber attacks against industrial control systems (ICS)

- A rise after Stuxnet in 2010 and high ever since
- New one always appears
  - Power outage in Ukraine(2015,2016), Operation Ghoul(2016), ...



Reference: NCCIC/ICS-CERT Year in Review

Background

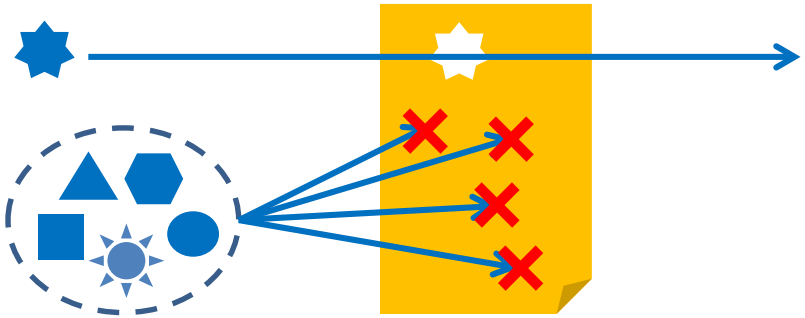
- ICS has its own requirements
  - Availability
    - Security must not slow nor stop the service
  - Long-term operation
    - Security must be effective throughout the lifetime

	General IT systems	ICS
Targets of security	Information	Facilities and Service
Priority of security	Confidentiality	*Availability*
Lifetime	3-5 years	*10-20 years*
Operating time	Business hours	*24 hours, 365 days*

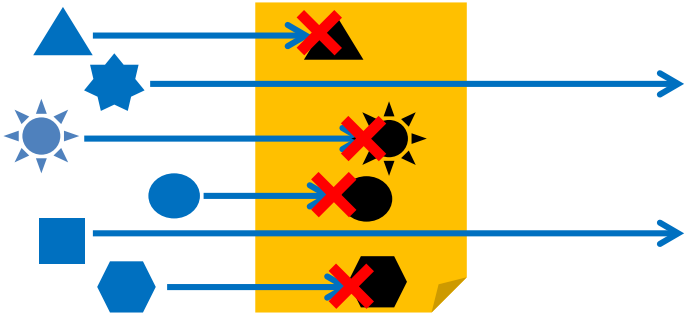
Whitelisting Network Monitor

- Whitelisting network monitor is suitable for ICS
  - Whitelist interprets deviations from the normal behavior as attacks

Whitelist
- List of "Allow", all the rest denied
- Potential to detect new attacks



Blacklist
- List of "Deny", all the rest allowed
- Accurate detection of known attacks



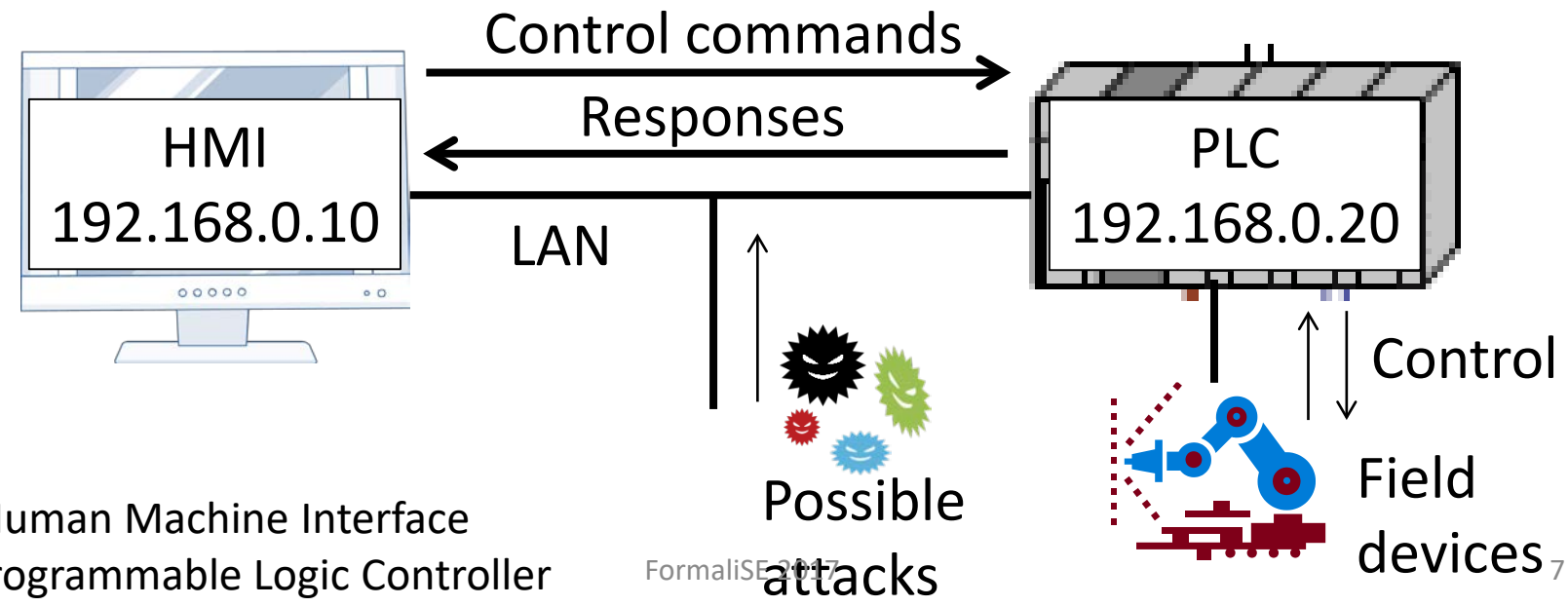
Whitelisting Network Monitor

- Why suitable for ICS?
  - Availability:
    - Pattern matching is lightweight
    - Whitelist doesn't need updating once defined
    - cf. antivirus software
  - Long-term operation:
    - Potential to detect new attacks in the future

Whitelisting Network Monitor

What's normal behavior?

- Consider a simple ICS
  - Consists of HMI, PLC and field devices
  - HMI/PLC sends commands/responses via LAN
  - Fixed "normal behavior" of commands/responses



HMI: Human Machine Interface
PLC: Programmable Logic Controller

Whitelisting Network Monitor

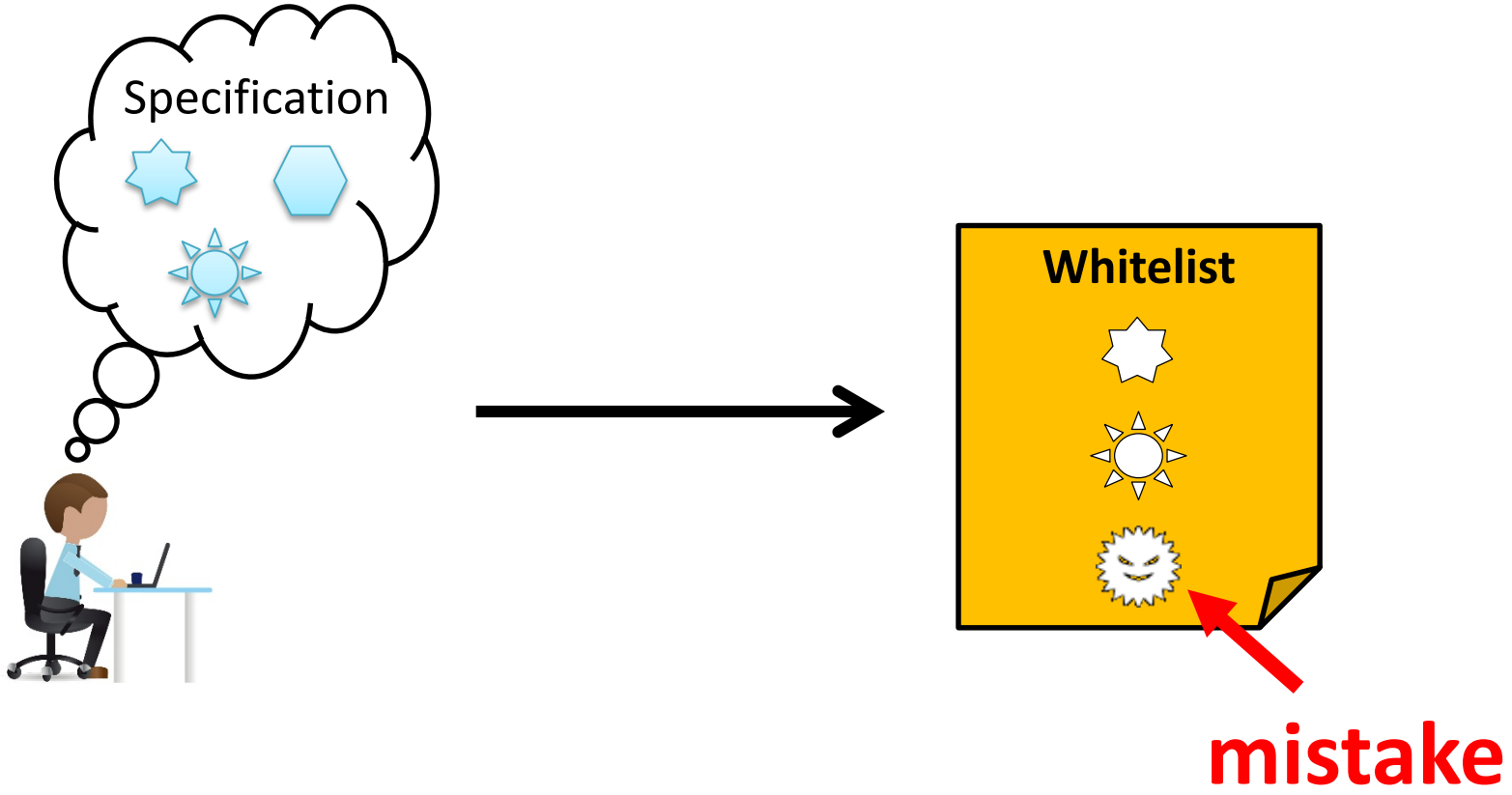
Normal behavior as a whitelist

State	Protocol	Sender	Receiver	Data length	Command	Payload condition	Period [ms]
Operation	Control	192.168.0.10 any	192.168.0.20 59306	1024	QuerySpeed	—	5
Operation	Control	192.168.0.10 any	192.168.0.20 59306	2048	ChangeSpeed	0〜90	50
…	…	…	…	…	…	…	…

Trusted Approach for Whitelisting

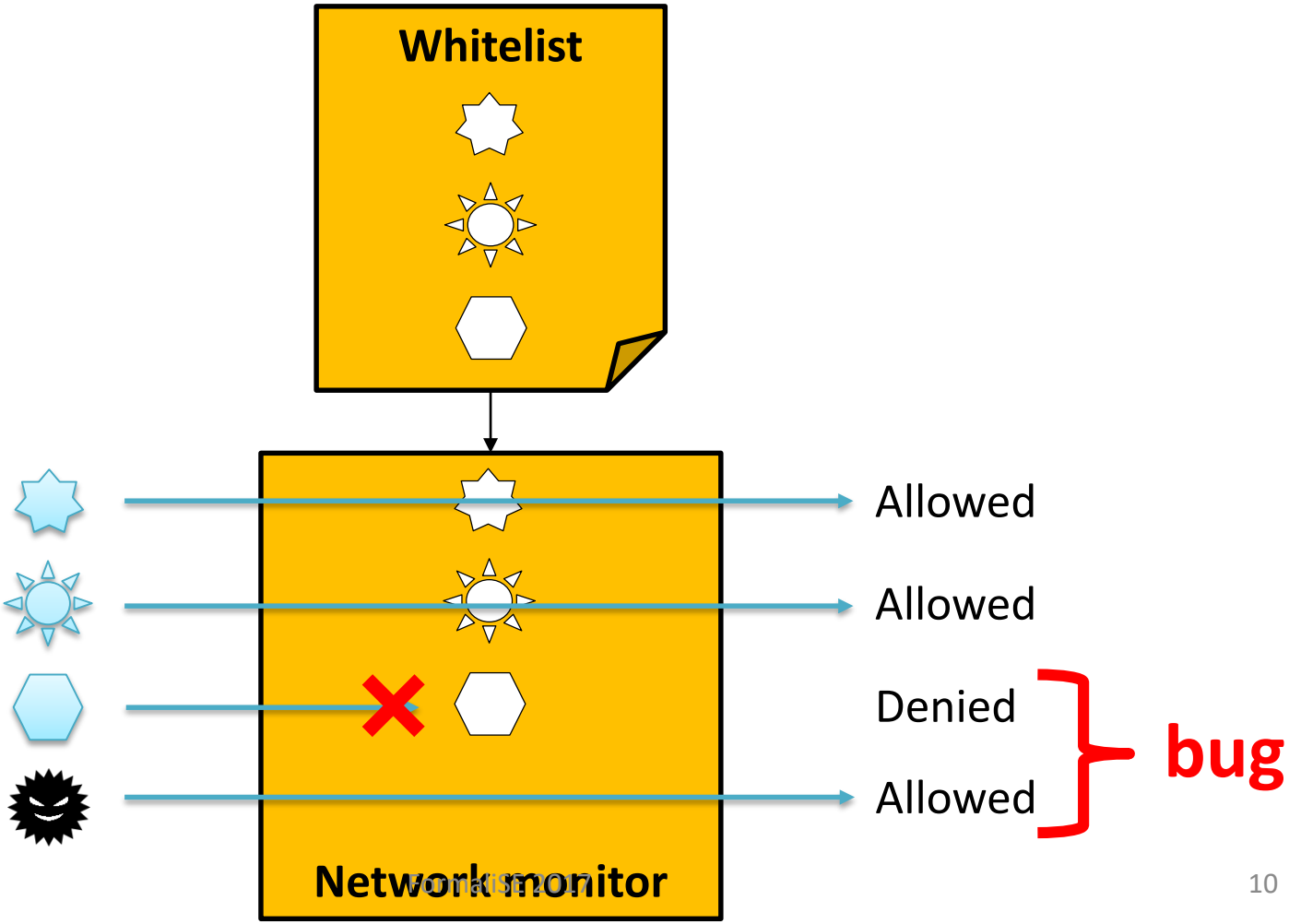
Motivation for a trusted approach

- A mistake in the whitelist causes a low detection rate or false positiveness



Trusted Approach for Whitelisting

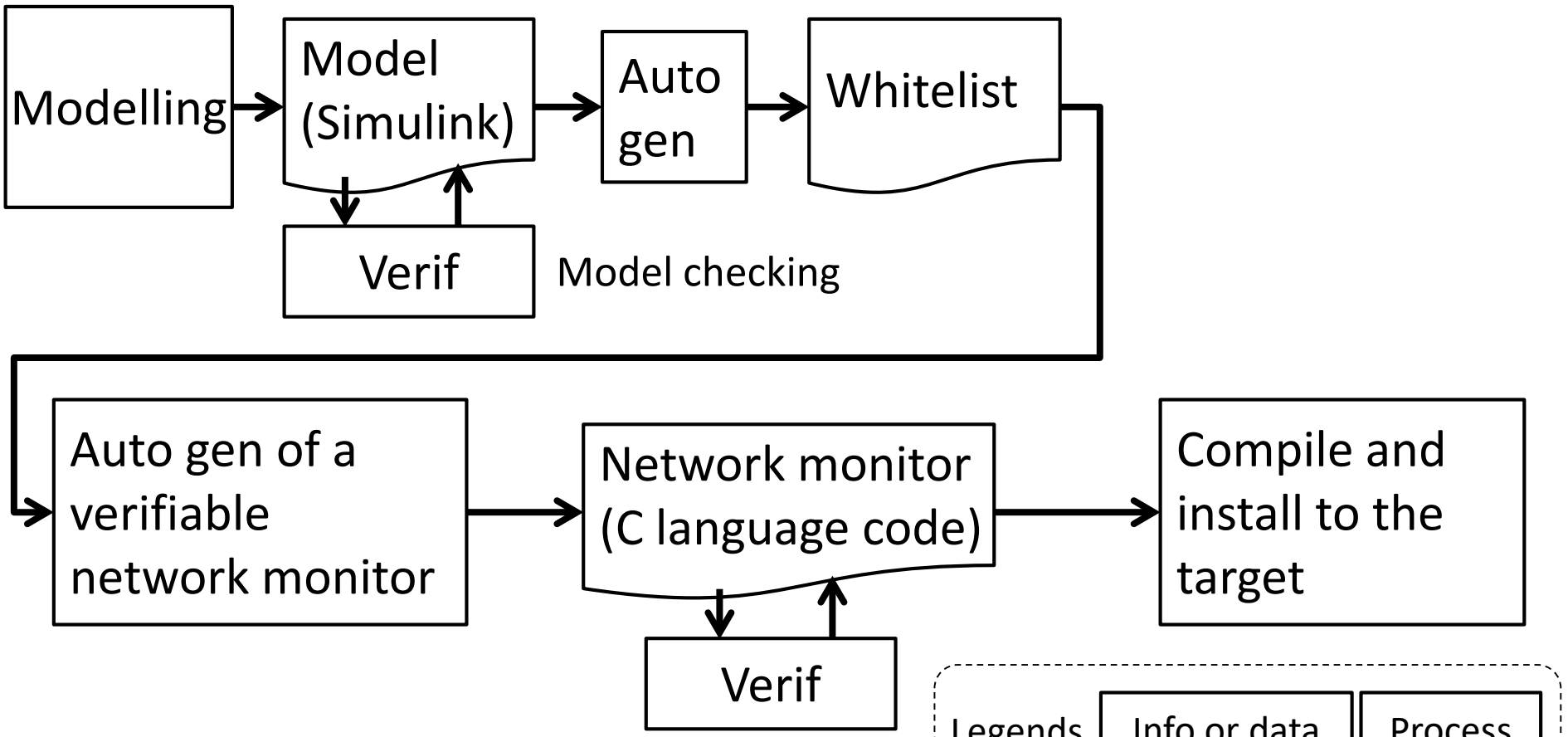
- A bug in the network monitor causes the same problem
- How to be confident about the network monitor?



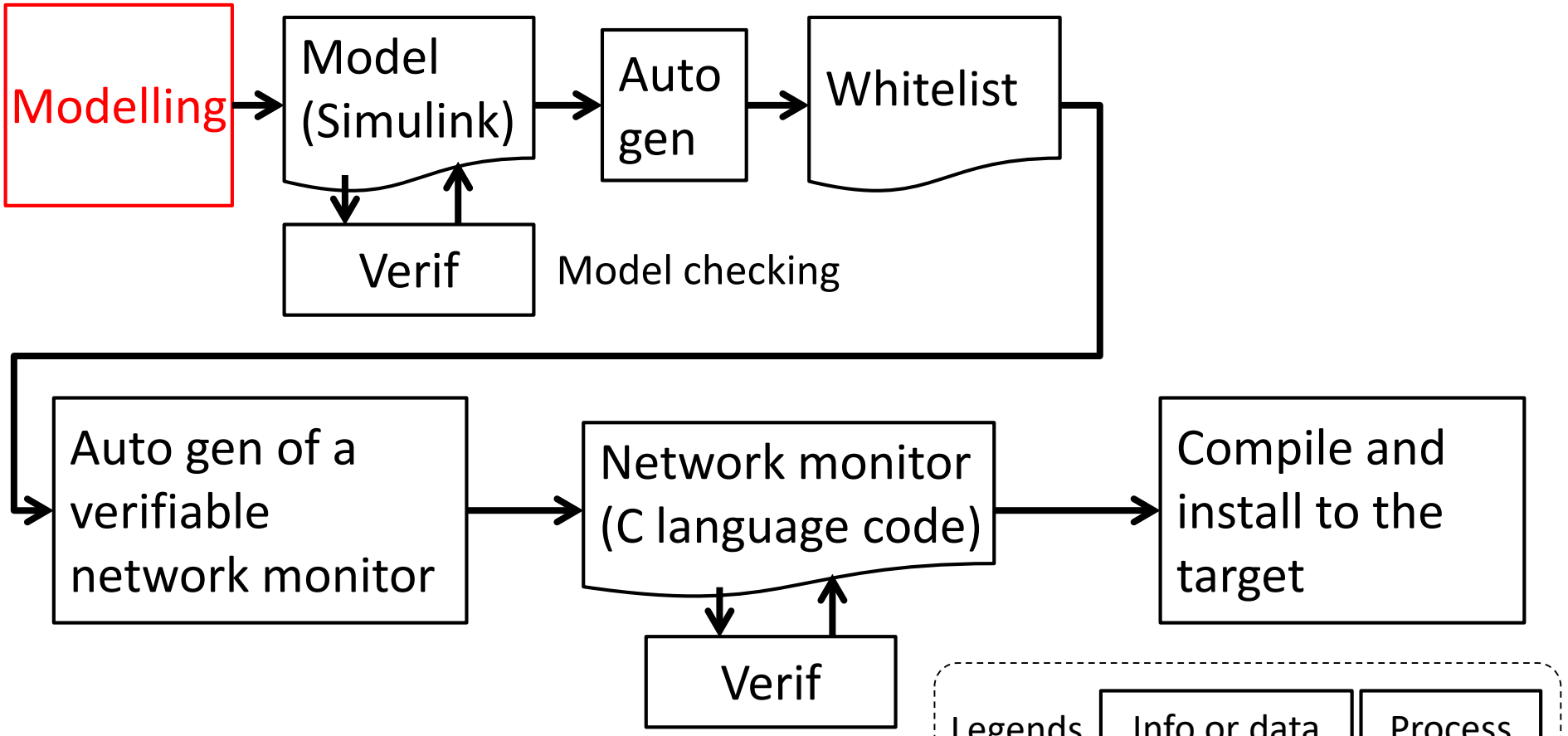
Trusted Approach for Whitelisting

- Proposal
  - Use of a model-based development framework
    - To clearly define a normal traffic specification
    - In the future, to be integrated into usual model-based development of industrial embedded software
  - Automation
    - The whitelist and the network monitor program are automatically generated to avoid manual mistakes
  - Verification
    - The model and the network monitor program are verified to ensure that there are no mistakes or bugs

Workflow



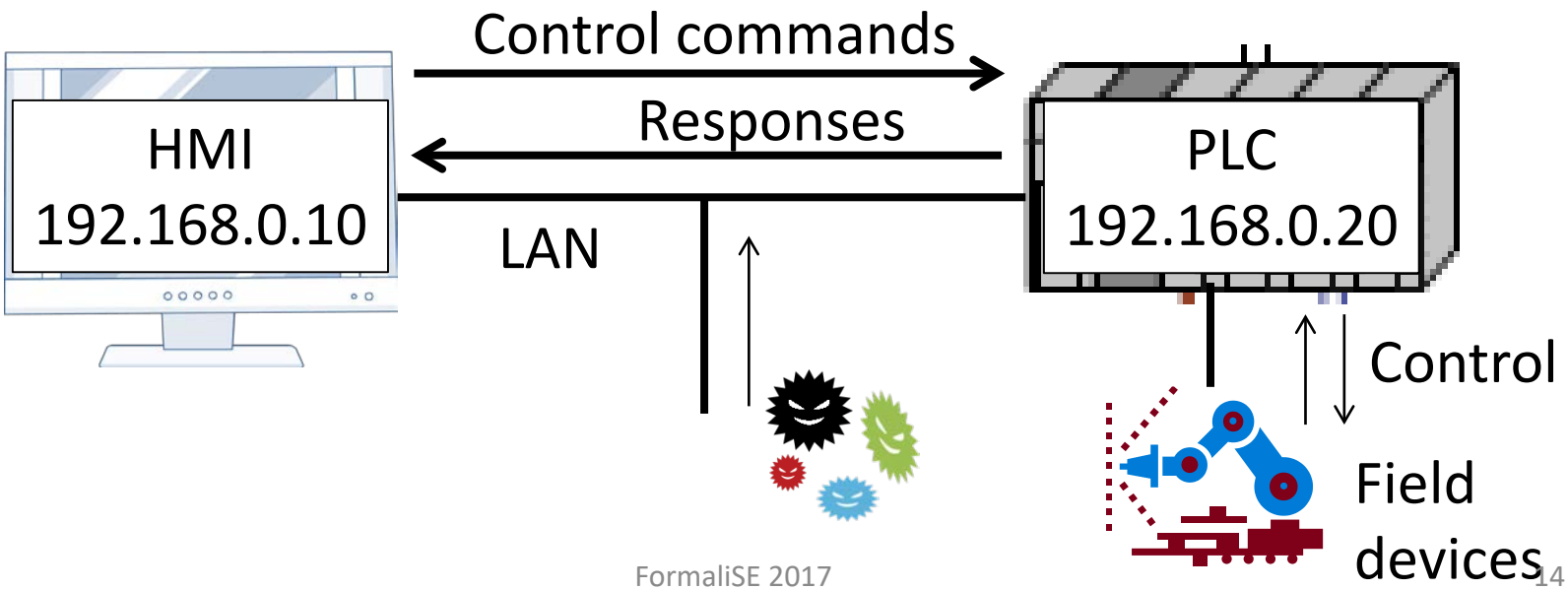
Workflow



Modelling of a control system

Toy example (again)

- Consists of HMI, PLC and field devices
  - but field devices are out of scope of modelling
- HMI/PLC sends commands/responses via LAN



Definition of system states and commands in use

- Just for ease
  - Three system states
  - Only "Operation" has a set of commands in use

System state	Command in use	Parameter	Period [ms]
Operation	ChageSpeed	0～90	50
	QuerySpeed	Device number	5
Maintenance	—	—	—
Abnormal	—	—	—

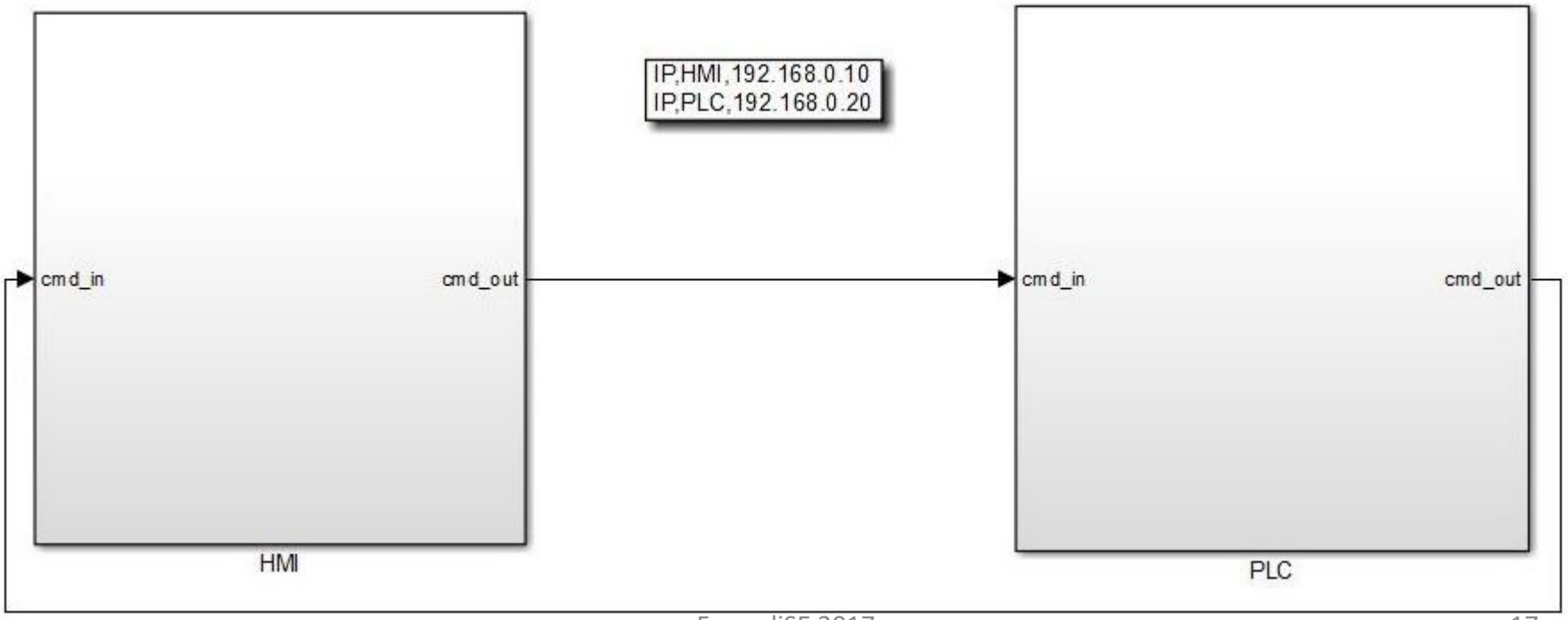
Modelling of a control system

Resulting whitelist (again)

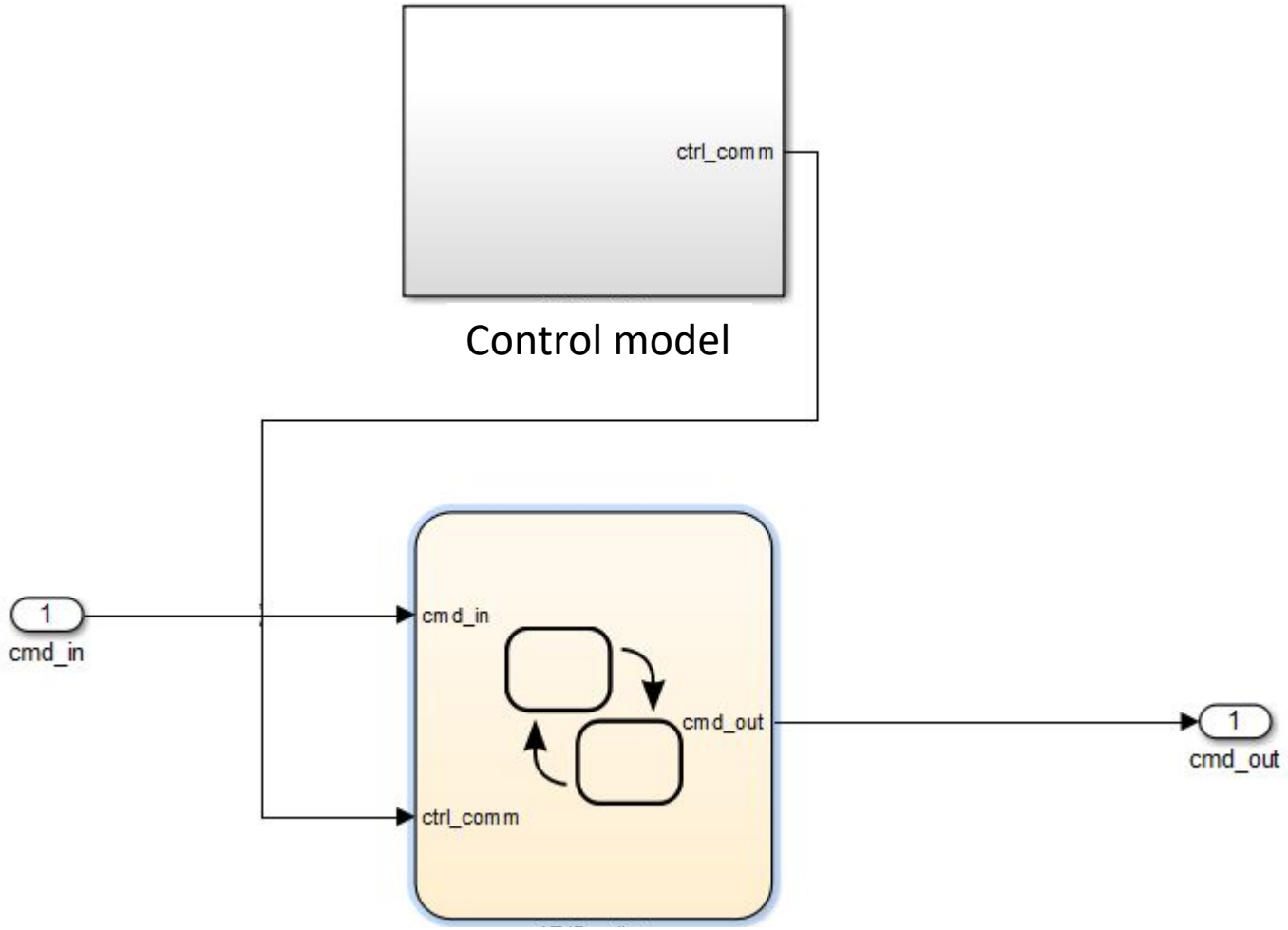
State	Protocol	Sender	Receiver	Data length	Command	Payload condition	Period [ms]
Operation	Control	192.168.0.10 any	192.168.0.20 59306	1024	QuerySpeed	Device number	5
Operation	Control	192.168.0.10 any	192.168.0.20 59306	2048	ChangeSpeed	0〜90	50
…	…	…	…	…	…	…	…

Modelling of a control system

Devices and connections between them at top level



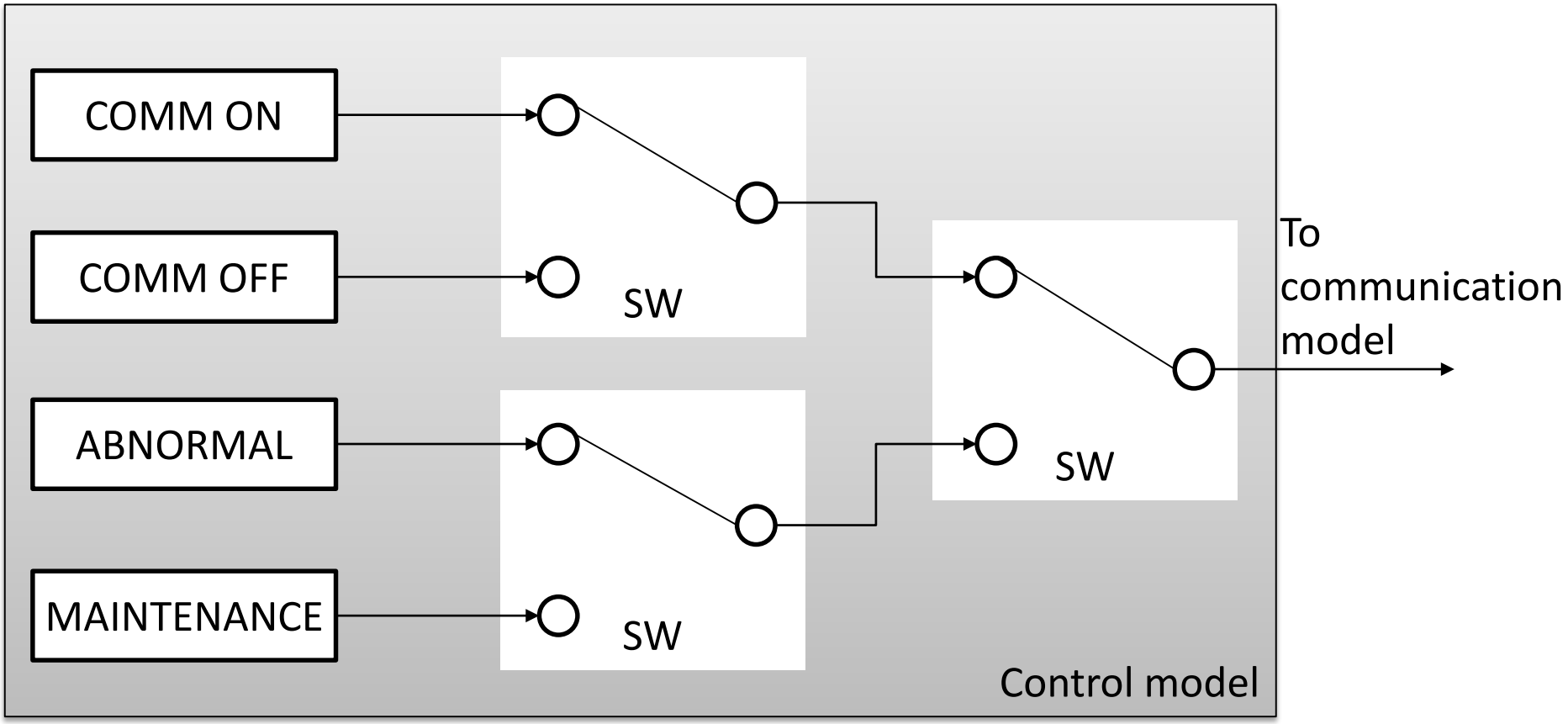
Control model and communication model inside each device



Modelling of a control system

Control model

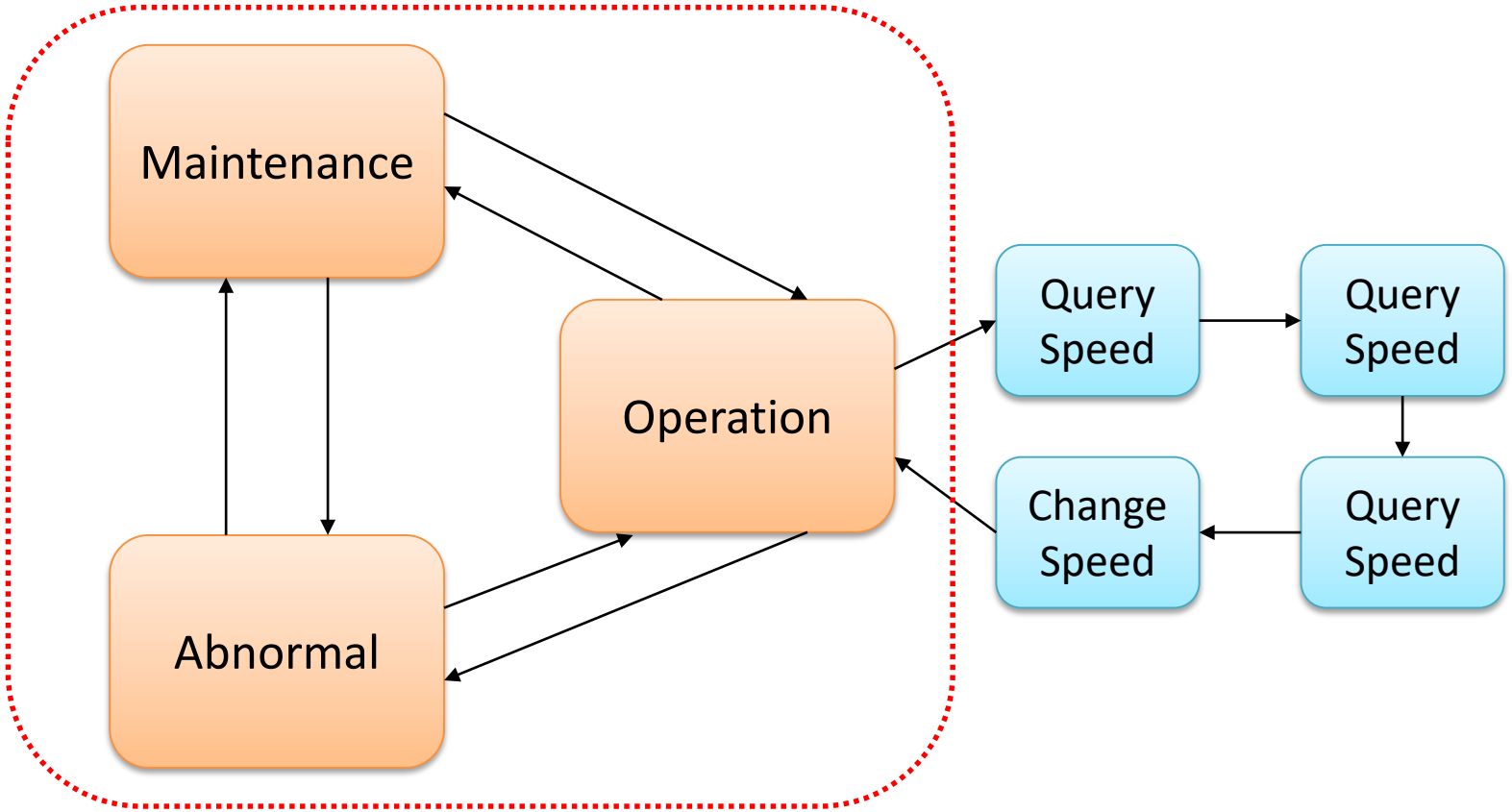
- Defines control behaviour
- Not needed for whitelist generation



Modelling of a control system

Communication model is a state machine that defines

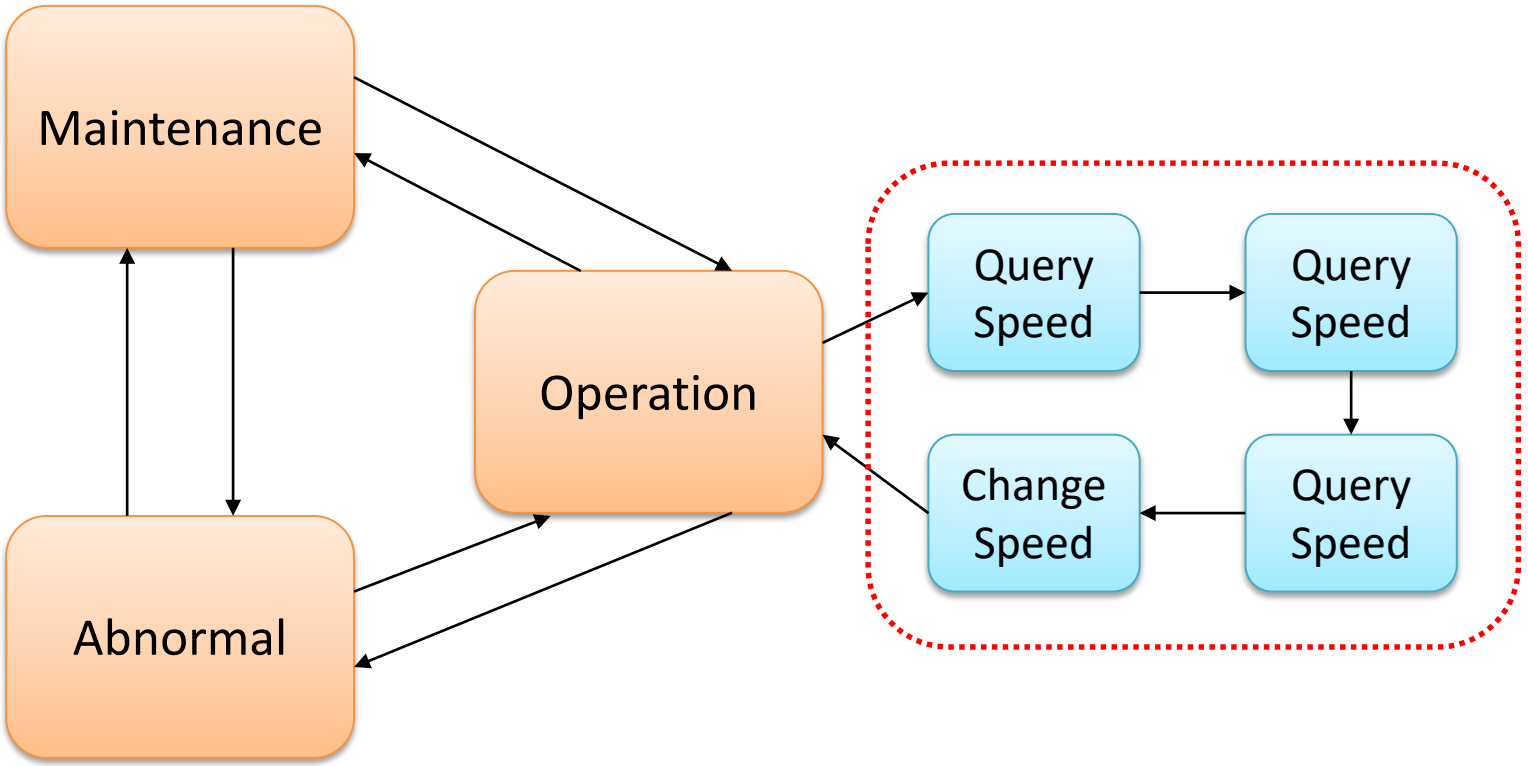
- System states and transitions between them
- Command sequence under each system state



Modelling of a control system

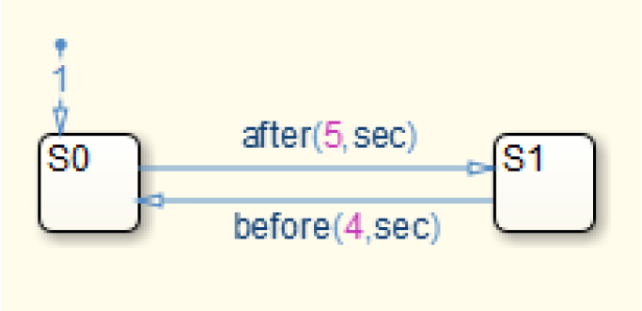
Communication model is a state machine that defines

- System states and transitions between them
- Command sequence under each system state

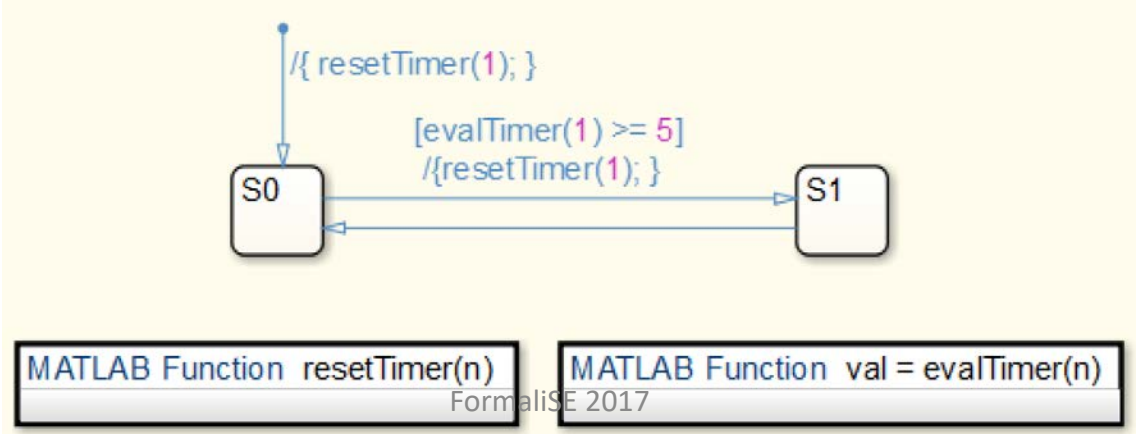


Modelling of a control system

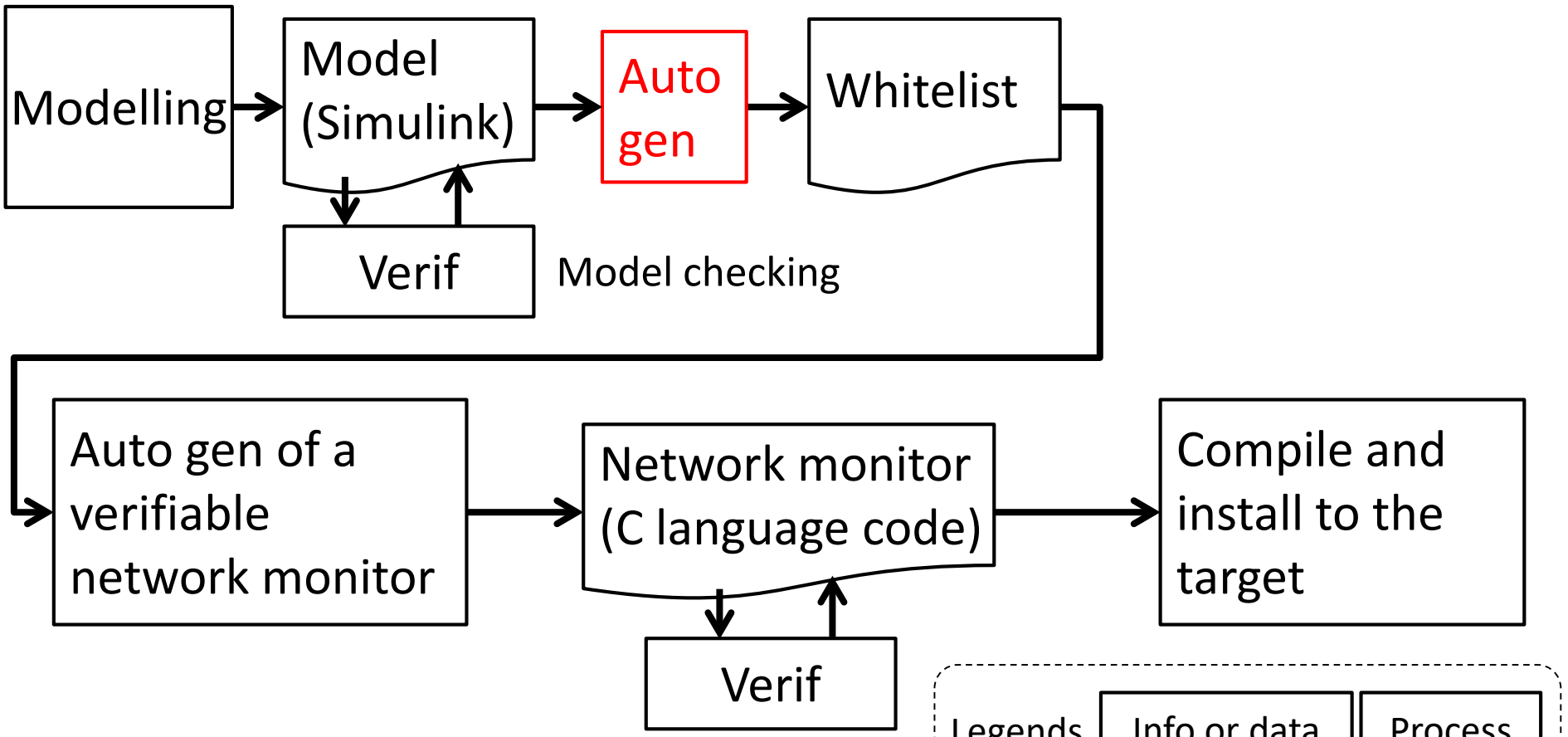
- Modelling of a period condition in Simulink
  - The guards "after" and "before" are close but not sufficient



  - Using global timers is too ad-hoc



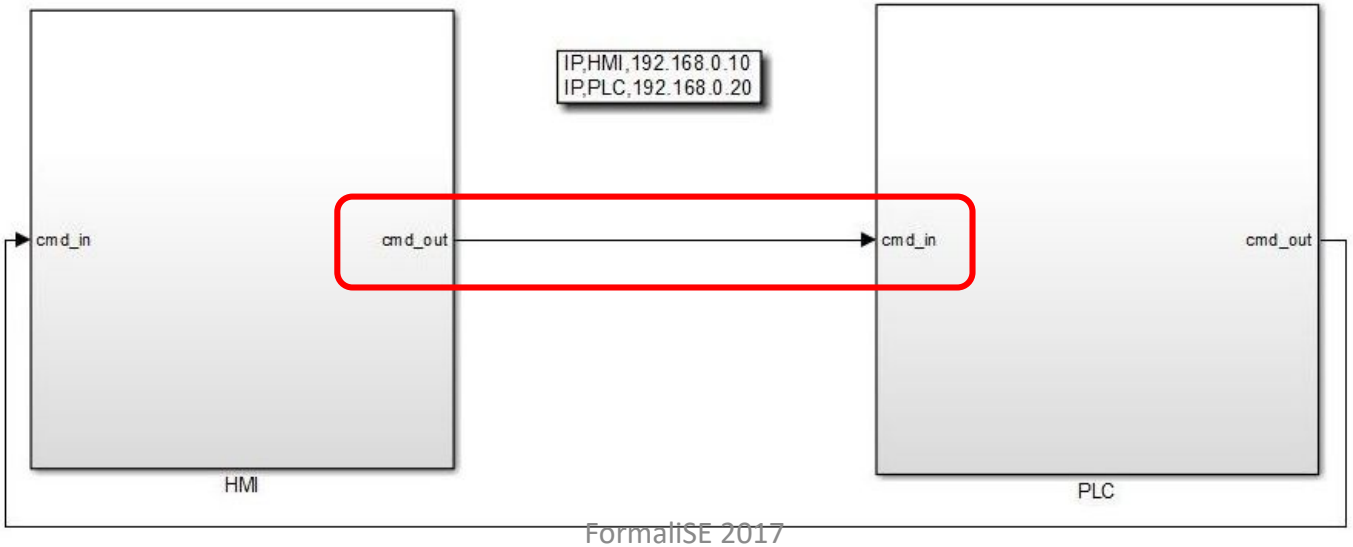
Workflow



Automated generation of the whitelist

- Extract command data from communication model
  - From cmd_out of HMI to cmd_in of PLC
  - Read1 is sent in the following example

```
Comm model on HMI side
  ... / {cmd_out.data = Read1; send(cmd_out)}

Comm model on PLC side
  cmd_in[cmd_in.data == Read1] / ...
```



Automated generation of the whitelist

Extracted data parsed, formatted and converted to whitelist

Extract

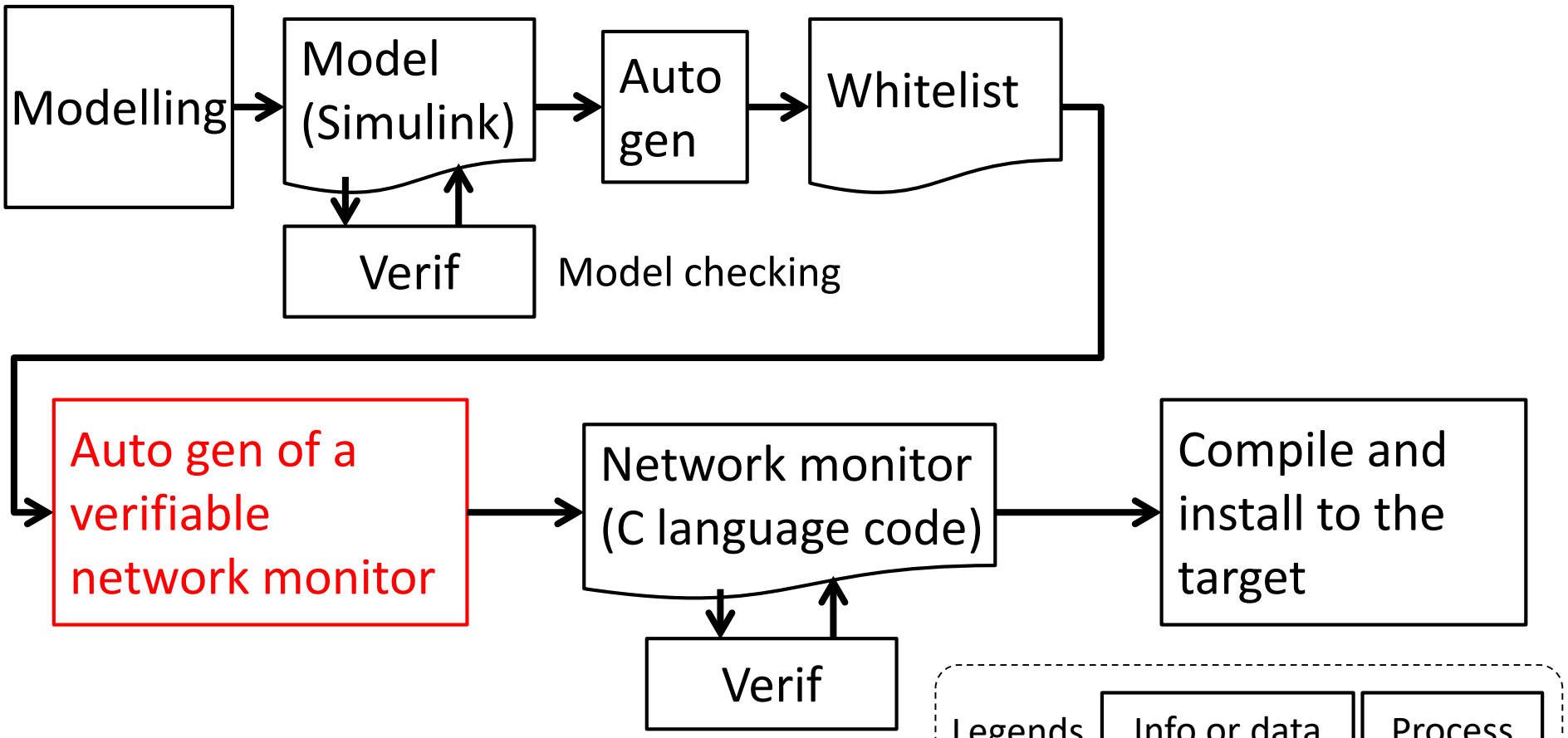
```
<Device num="2">
HMI
PLC
</Device>

<IP num="2">
IP,HMI,192.168.0.10
IP,PLC,192.168.0.20
</IP>

<Communication
device="HMI">

<State num="26">
74:SYS_Maintenance
75:SYS_Operation
...
```

Parse & Format

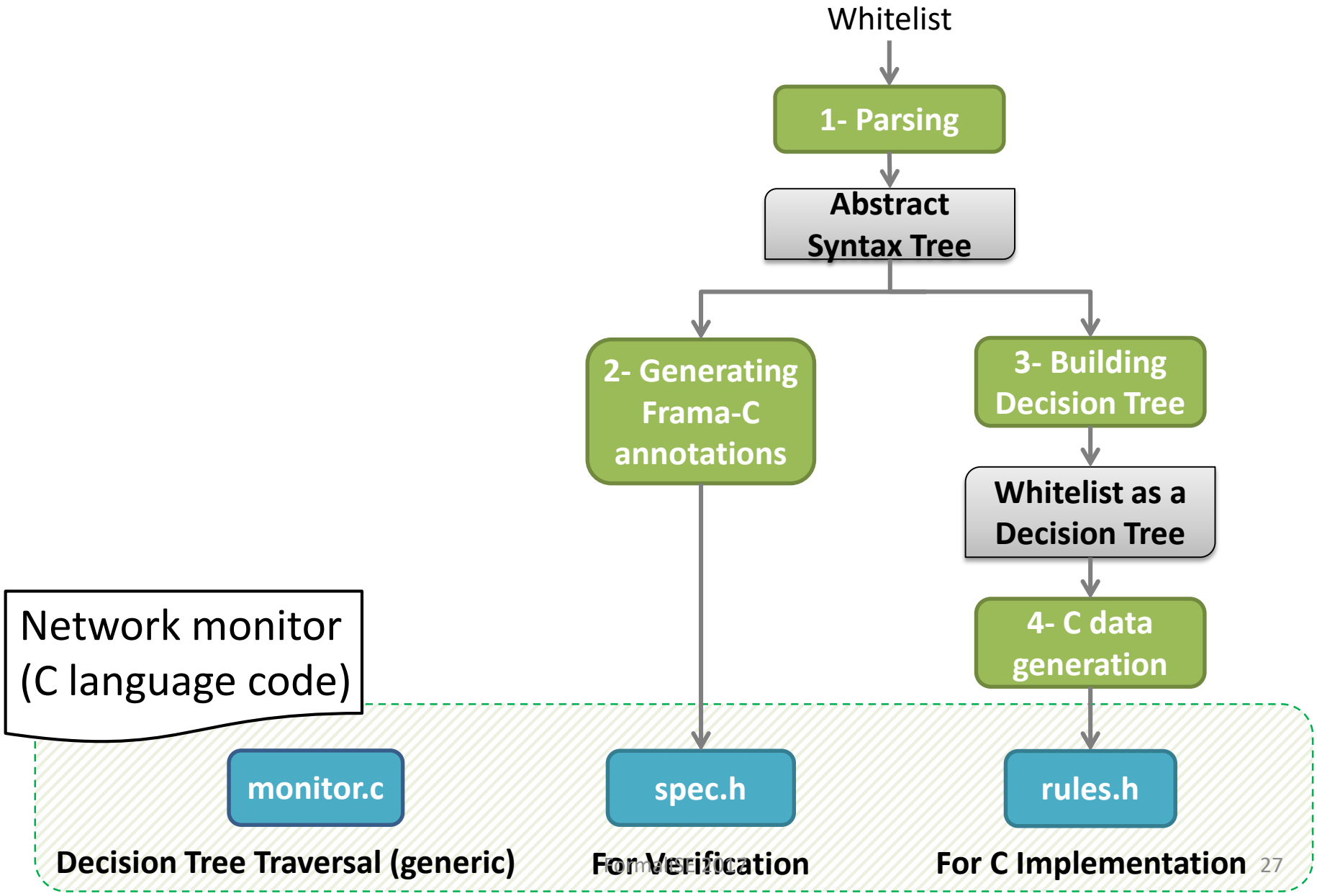
```
HMI, PLC, SYS_Operation, Read1
HMI, PLC, SYS_Operation, Read2
HMI, PLC, SYS_Operation, Read2
HMI, PLC, SYS_Operation, Read2
HMI, PLC, SYS_Operation, Read2
HMI, PLC, SYS_Operation, Read2
HMI, PLC, SYS_Operation, Read2
HMI, PLC, SYS_Operation, Read2
HMI, PLC, SYS_Operation, Read2
HMI, PLC, SYS_Operation, Read2
HMI, PLC, SYS_Operation, Read2
HMI, PLC, SYS_Operation, Read2
HMI, PLC, SYS_Operation, Read2
HMI, PLC, SYS_Operation, Read2
```

Conversion to whitelist using detailed information at implementation level

Workflow



Automated generation of the network monitor



Automated generation of the network monitor

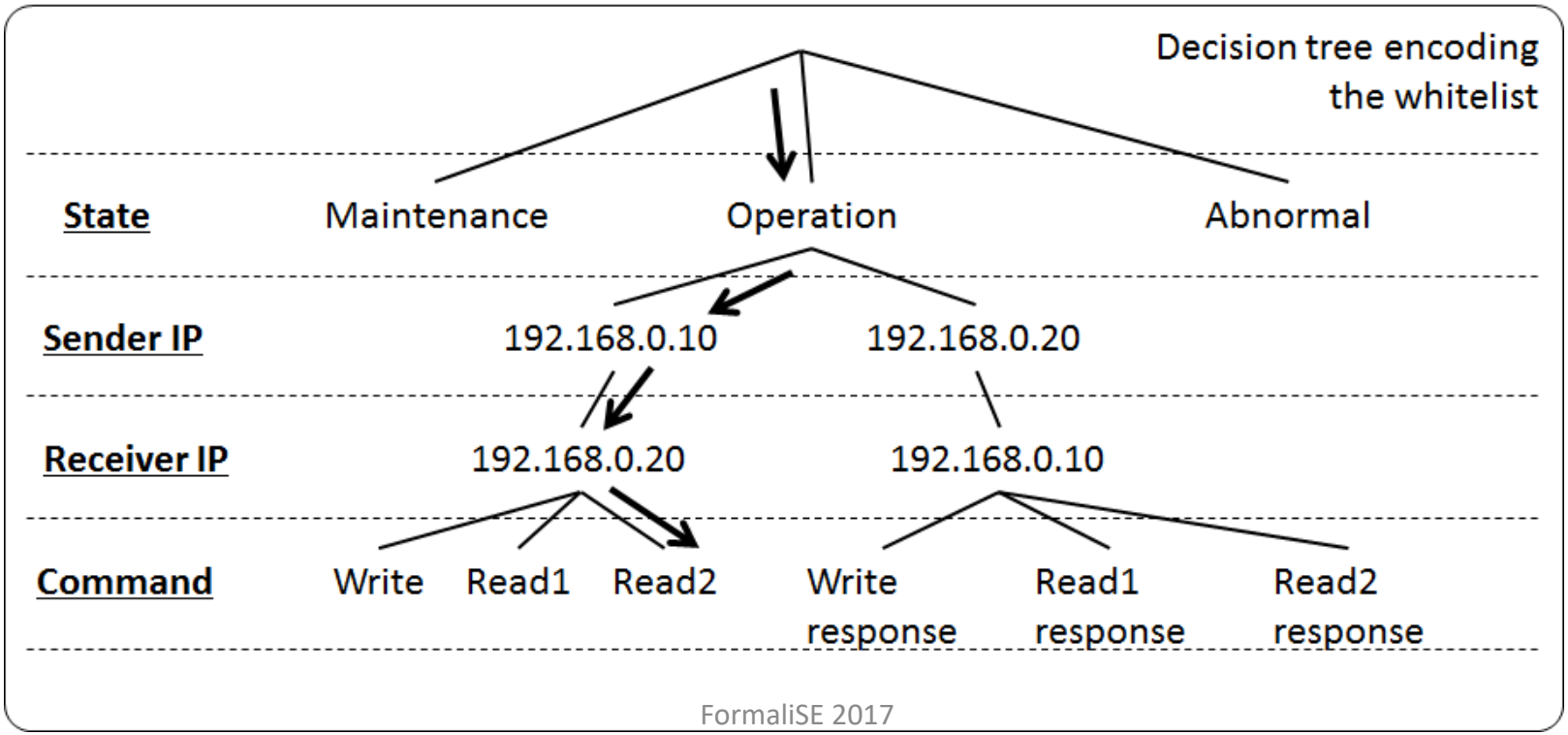
Whitelisting as a decision tree

System state

State
Operation

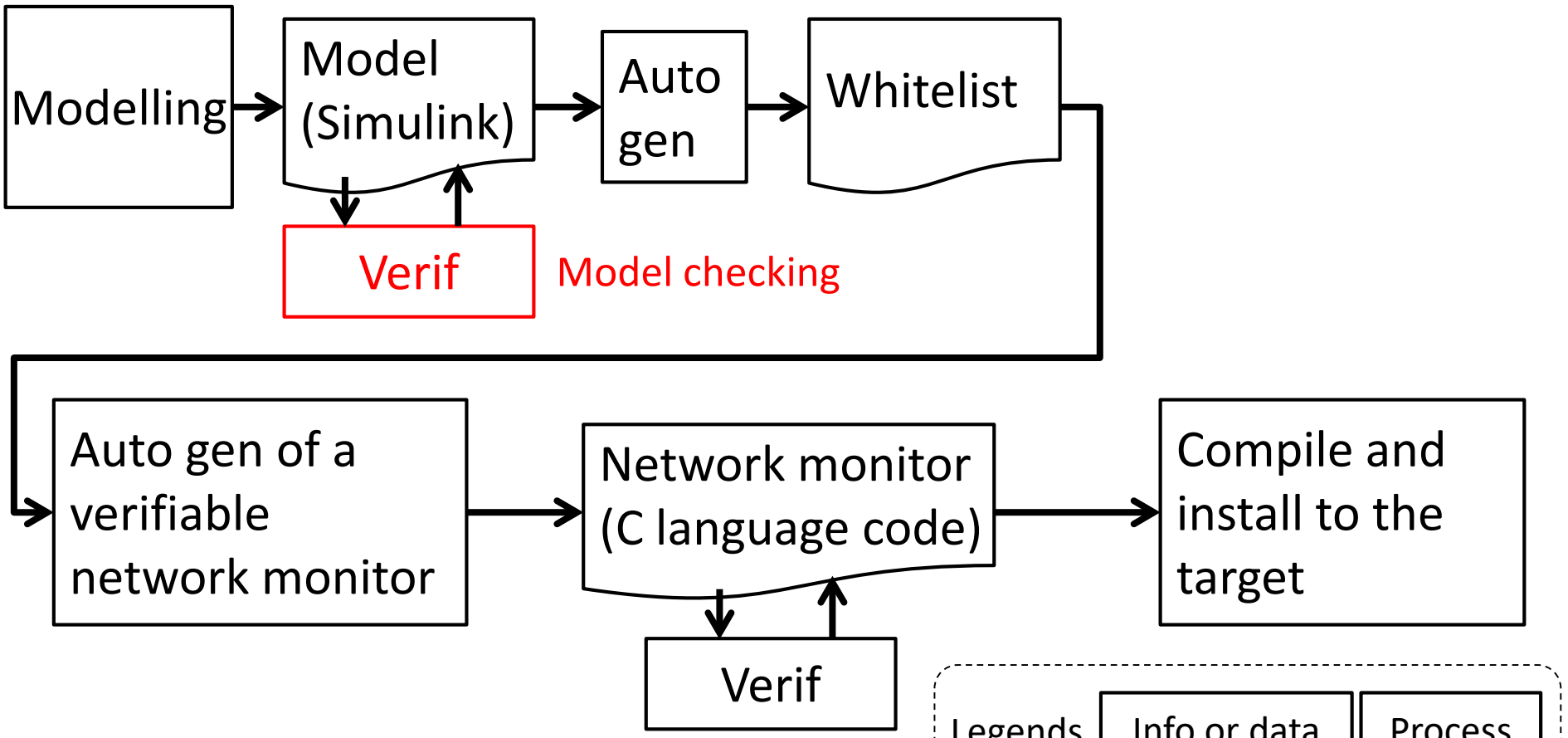
Network packet data

Sender IP	Receiver IP	Command
192.168.0.10	192.168.0.20	Read2



Verification of the model

The model is the starting point of the trusted monitor, so needs verification by model checking

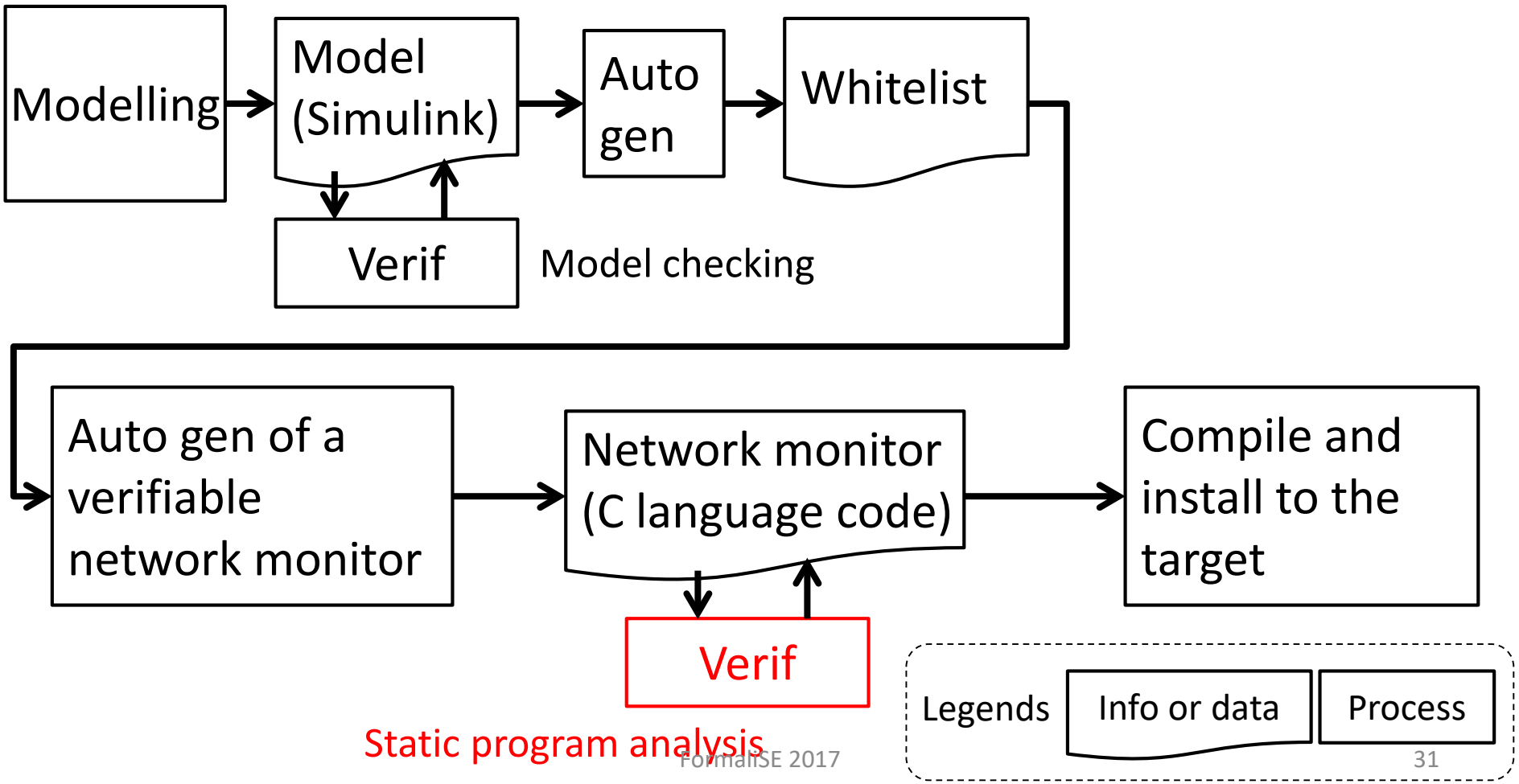


Verification of the model

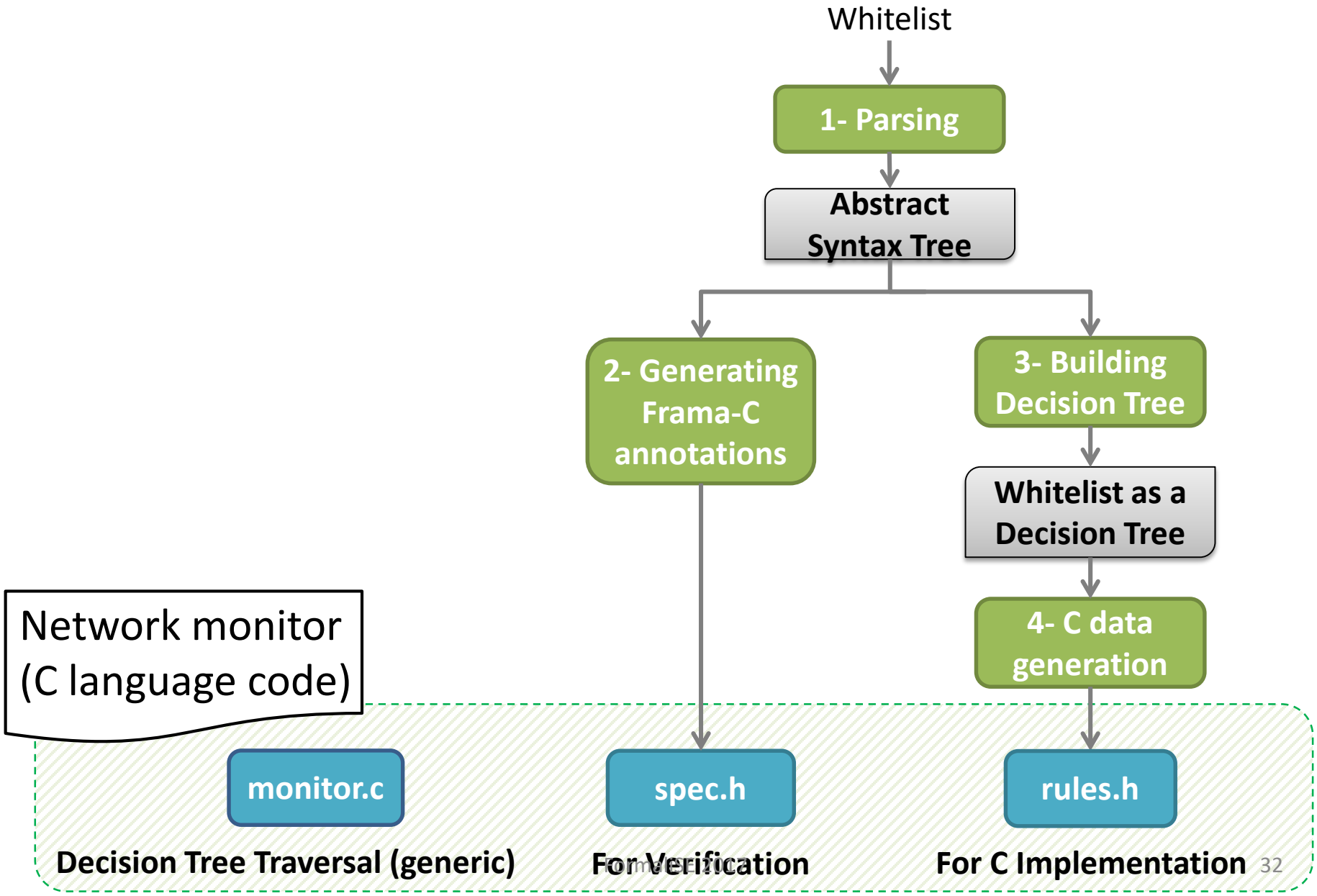
- Experiment using Simulink Design Verifier (SLDV)
  - Run-time errors and dead logics can be detected
    - The latter is of some use
  - Proof of properties defined in Simulink
    - Takes a lot of time for a model that has a relatively large state machine of our purpose
- As a result, SLDV is not sufficient for our purpose

Verification of the network monitor

The network monitor is responsible for detection and must be free of bugs



Verification of the network monitor



Verification of the network monitor

- Frama-C/WP
  - Static analysis and verification of C programs
  - The WP plugin for formal reasoning
  - The following example asserts
    - i < MAX_INT to avoid integer overflow
    - the pointer "data" is valid

```
//@assert signed_overflow: i+1<=2147483647;
i++;
//@assert mem_access: ¥valid_read(data);
if (*data == 1) {
...
}
```

Verification of the network monitor

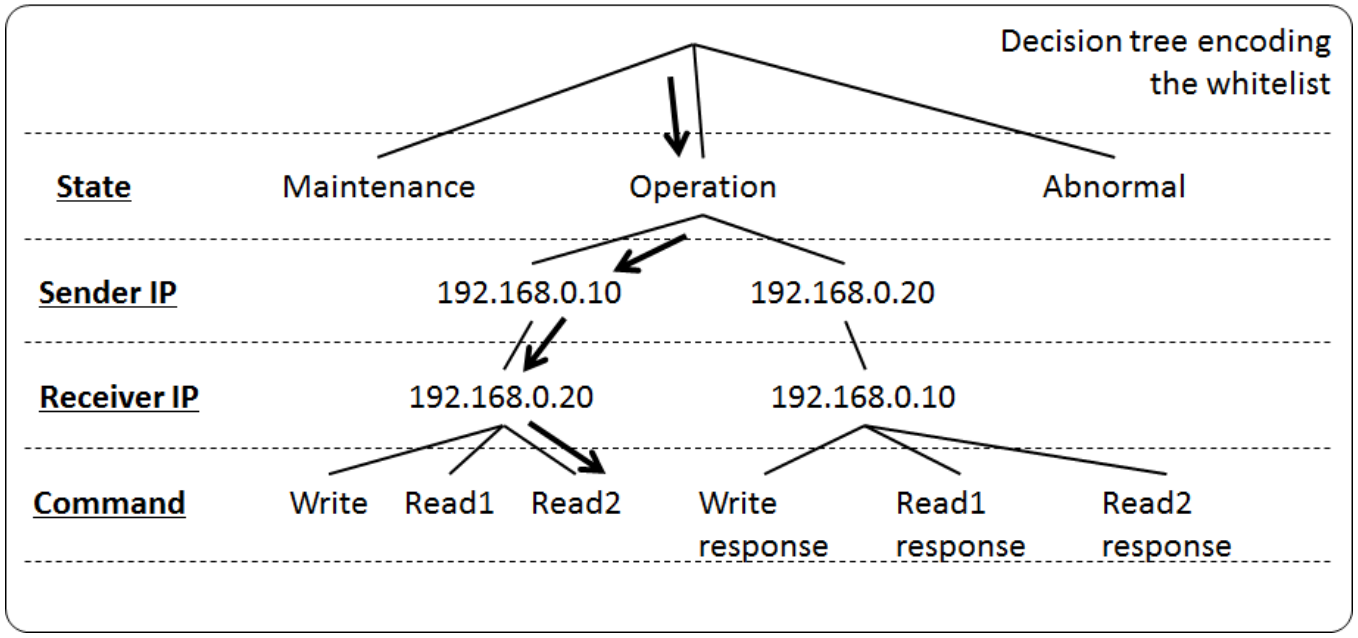
Whitelisting as a decision tree

System state

State
Operation

Network packet data

Sender IP	Receiver IP	Command
192.168.0.10	192.168.0.20	Read2



Verification of the network monitor

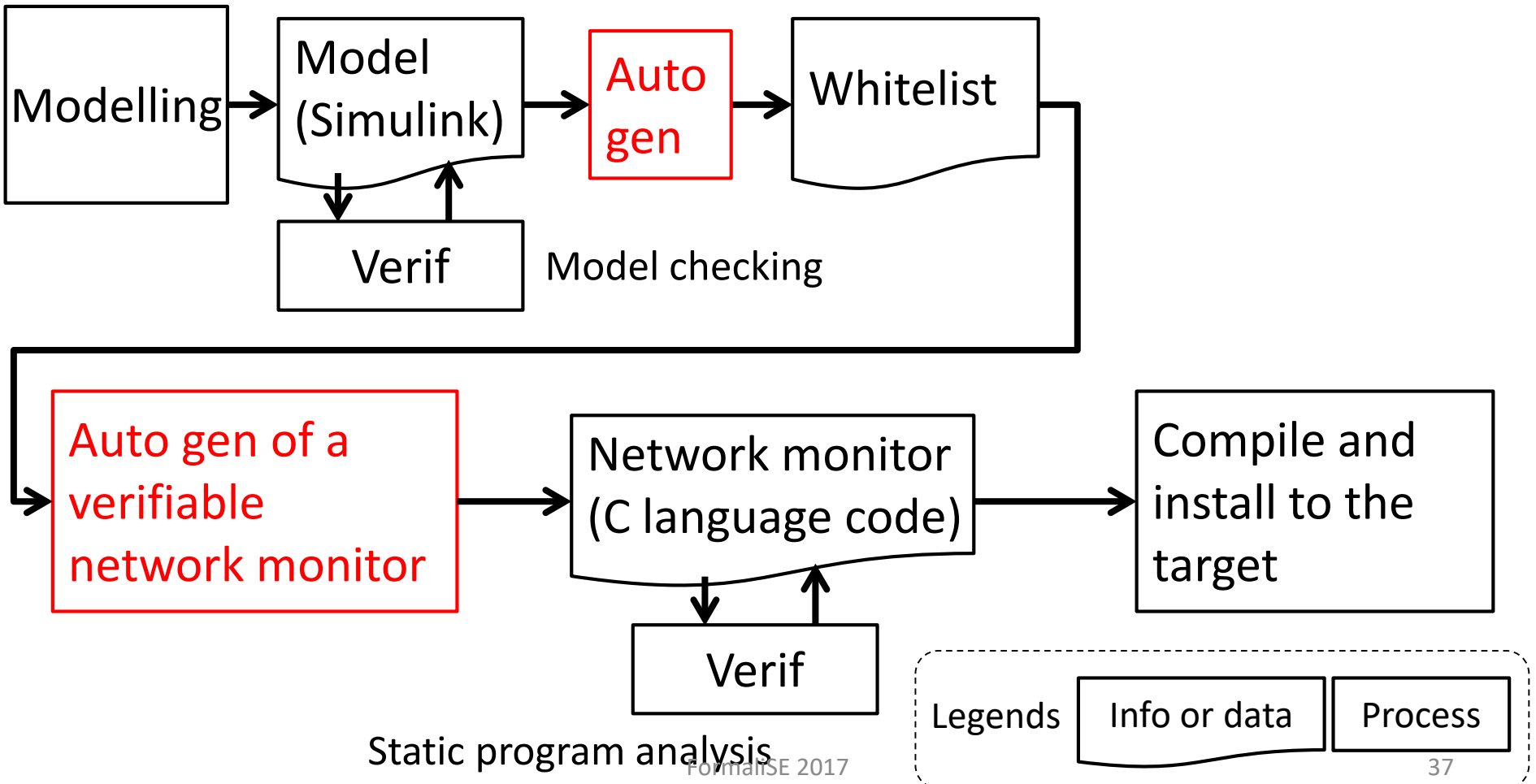
```
/*@requires ¥valid((tree_t*) rules)
           && valid_tree_t(*rules);
  @requires parsed;
  @ensures ¥result == 1 ==> matched;
  @assigns ¥nothing; */
int monitor(void);
```

Verification of the network monitor

```
@predicate rule_2 =
  state == 0                          &&
  send_info[IP] == 0x0a80001          &&
  0 <= send_info[TCP] <= 65535        &&
  recv_info[IP] == 0xc0a80014         &&
  recv_info[TCP] == 0x0c              &&
  command == 0x2112                   &&
  timers[0] == 5;
```

Unverified parts of the workflow

No verification about conversion is not a big issue



Conclusion

- Proposal
  - Trusted whitelisting network monitor
    - Automated generation of the whitelist based on a model-based development framework, where the model can be verified
    - The network monitor is automatically generated from the whitelist and can be proven to be free of bugs
  - Open problems / future works
    - Modelling: period conditions can't be encoded but are supposed to be given as an implementation detail
    - Automation: whitelist generation is not fully automated
    - Verification: model verification needs a different tool
    - Others: evaluation of the detection rate, etc.