Design for Dependability and its Challenges

IEEE ETR Round Table 2024

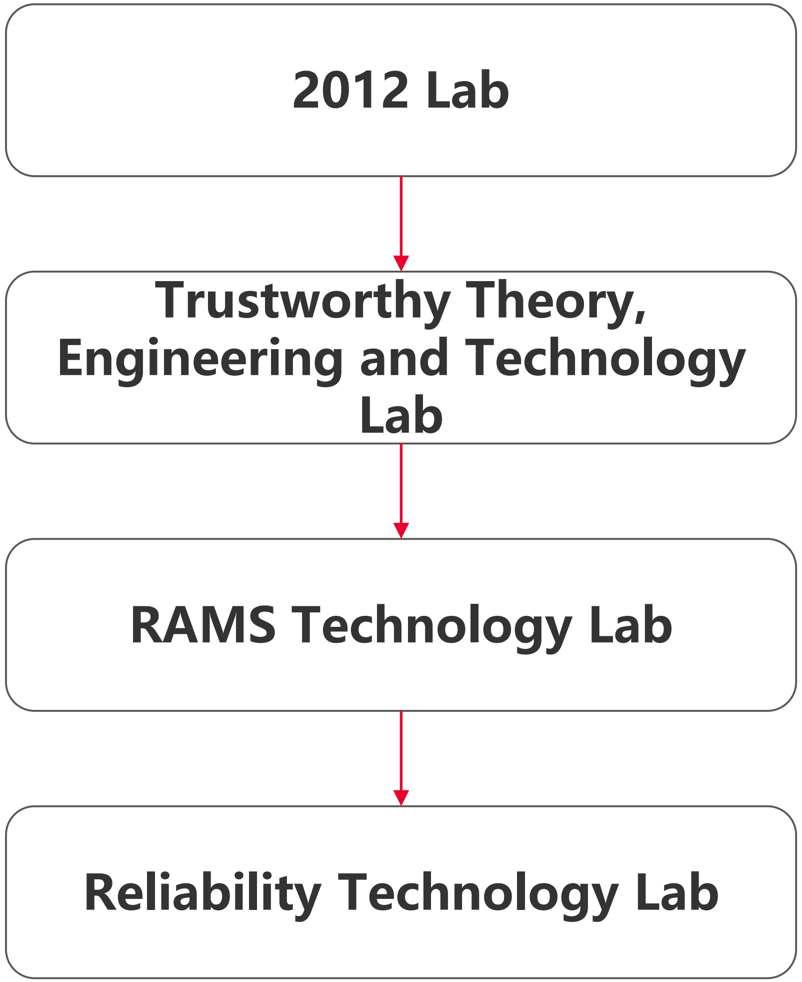
Name: Chengqiang Huang (Vincent)
Date: 2024.05.21

HUAWEI

Contents

1. Introduction
2. Design for Dependability
3. Model Verification, Validation, and Runtime Assurance
4. Existing Challenges

1. Introduction：Reliability Technology Lab



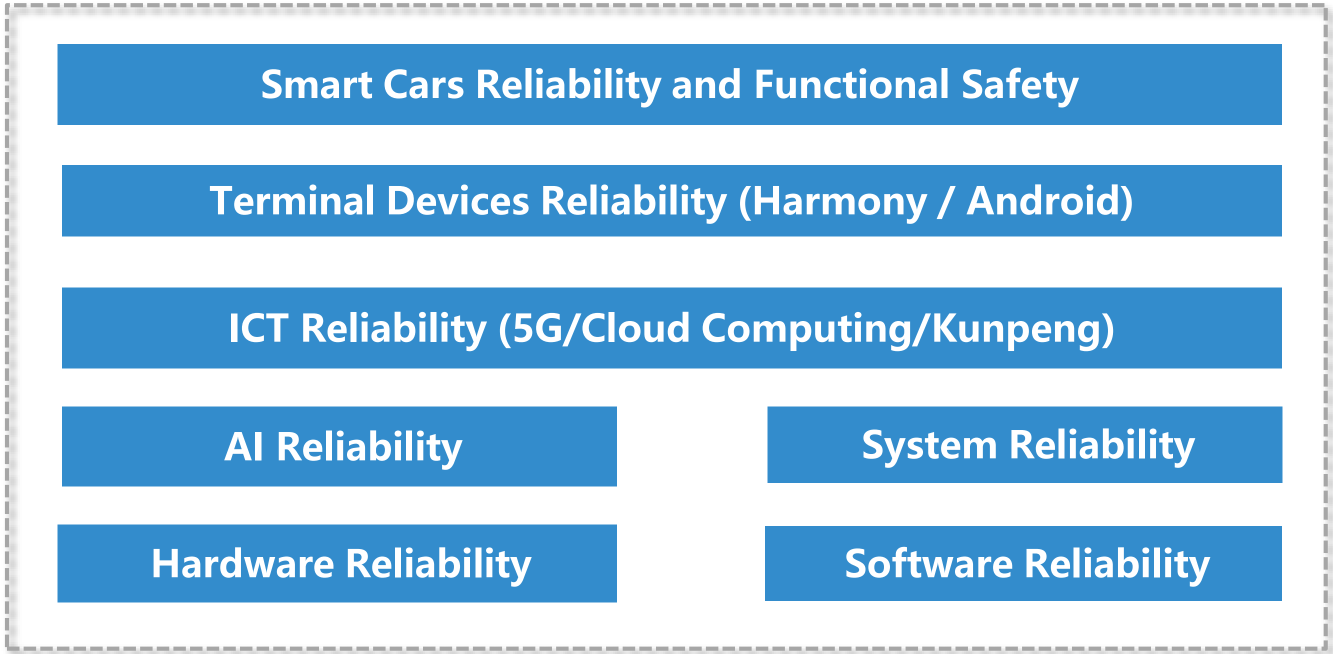
2012 Lab

Trustworthy Theory, Engineering and Technology Lab

RAMS Technology Lab

Reliability Technology Lab

Reliability, Availability, Safety



Smart Cars Reliability and Functional Safety

Terminal Devices Reliability (Harmony / Android)

ICT Reliability (5G/Cloud Computing/Kunpeng)

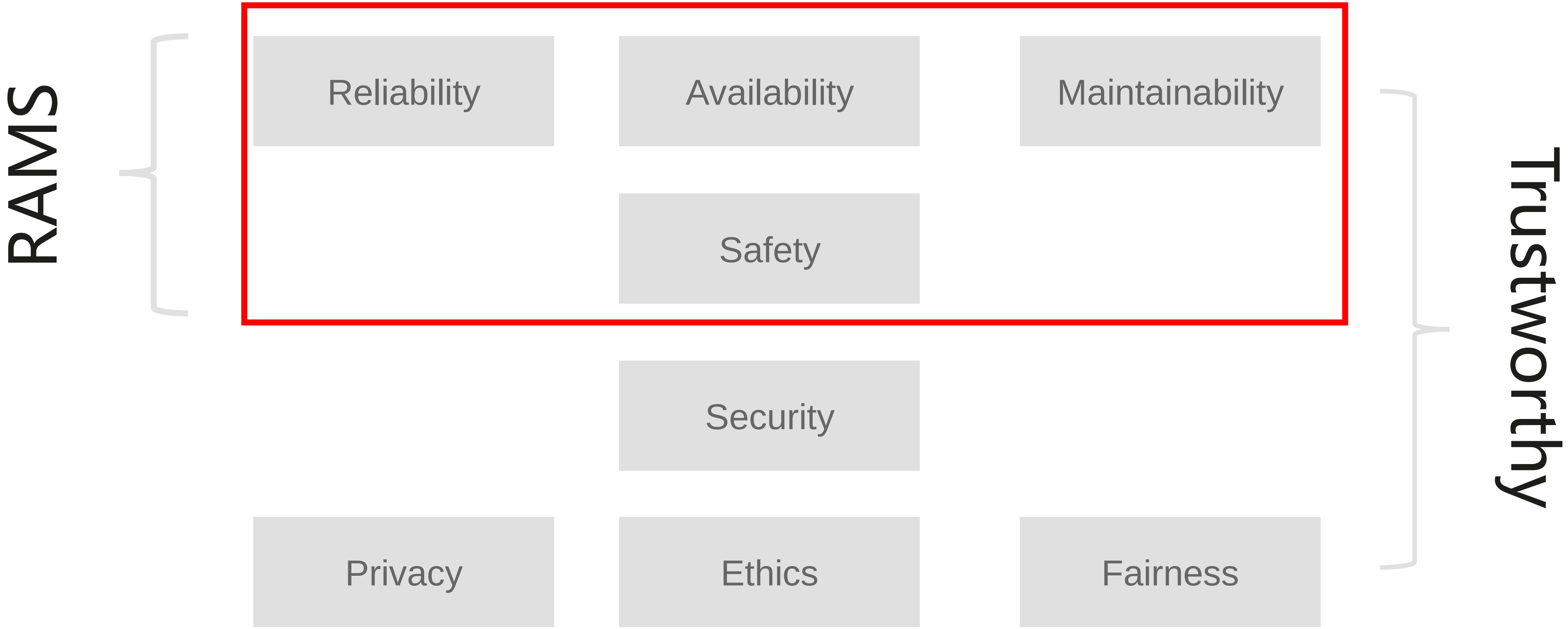
AI Reliability

System Reliability

Hardware Reliability

Software Reliability

1. Introduction：The scope



RAMS

Reliability

Availability

Maintainability

Safety

Security

Privacy

Ethics

Fairness

Trustworthy

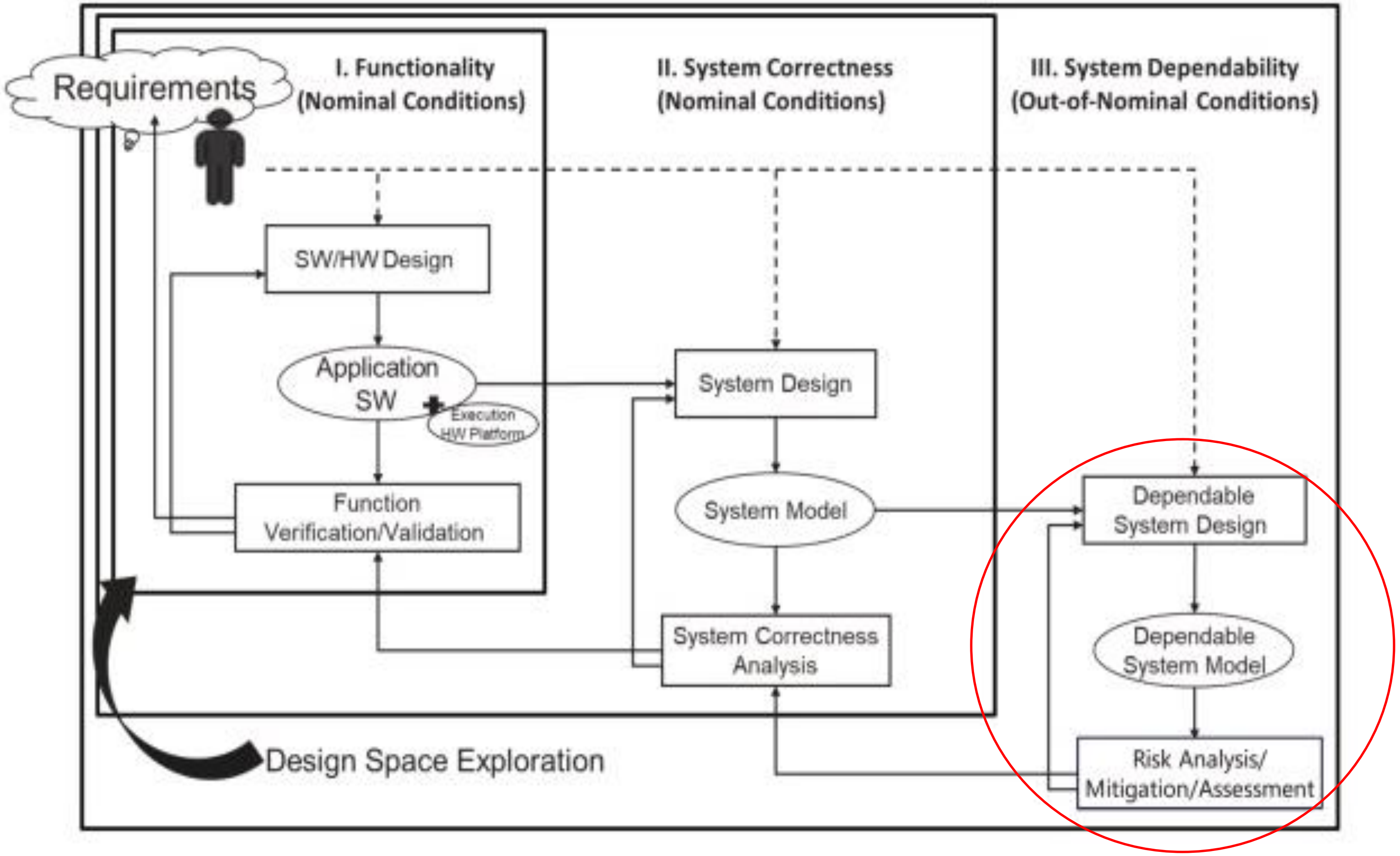
Cf. fortiss:Trustworthy Autonomous/Cognitive Systems

2. Design for Dependability： the Methodology

Design for dependability — State of the art and trends ☆

Pub date: 5 February 2024

Hezhen Liu [a], Chengqiang Huang [a], Ke Sun [a], Jiacheng Yin [a], Xiaoyu Wu [a], Jin Wang [a], Qunli Zhang [a], Yang Zheng [a], Vivek Nigam [b], Feng Liu [b], Joseph Sifakis [c]

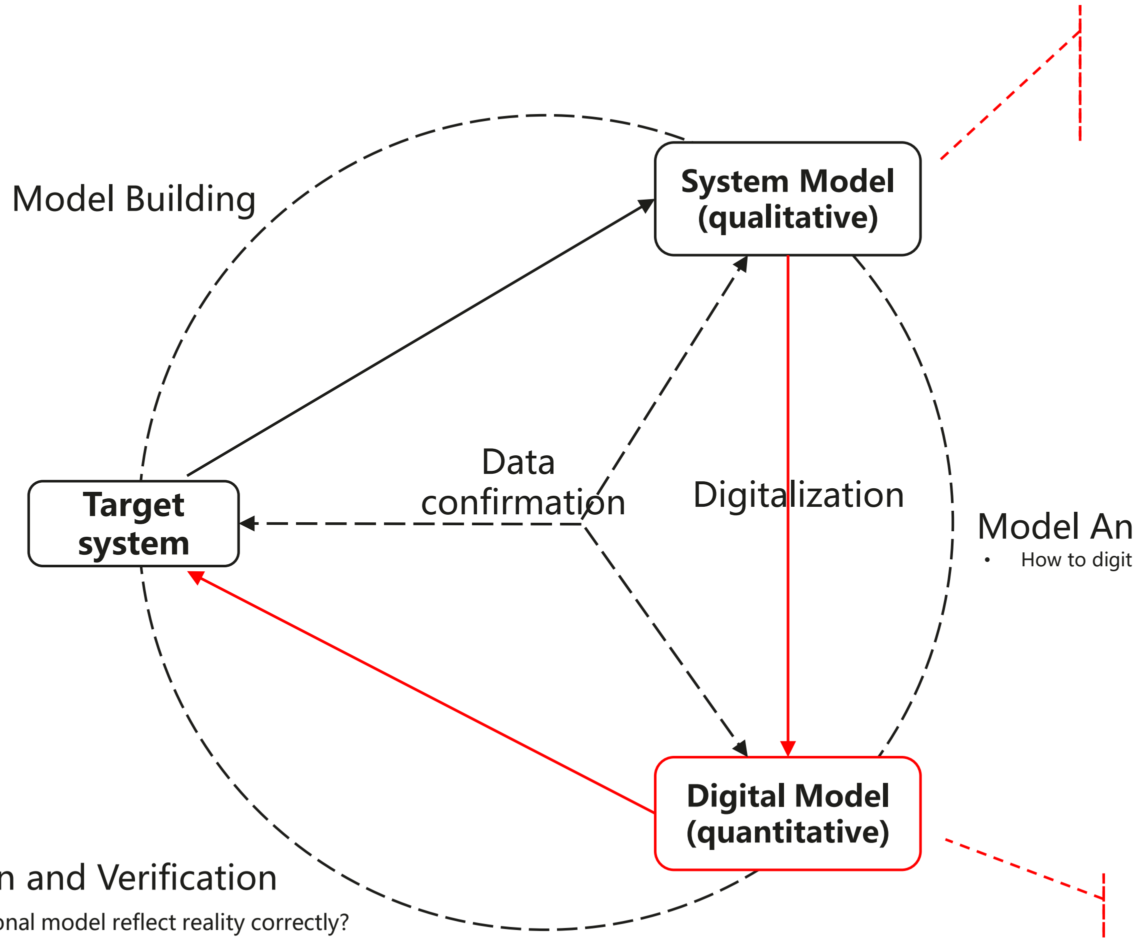


The third block emphasize the importance of dependability in system design with the consideration of out-of-nominal conditions troughout the system lifecycle.

Risk analysis, mitigation and assessment are three key techniques to promote system dependability.

Liu, H., Huang, C., Sun, K., Yin, J., Wu, X., Wang, J., ... & Sifakis, J. (2024). Design for dependability—State of the art and trends. *Journal of Systems and Software*, 111989.

2. Design for Dependability： the Models



Recorded in docs, ppts, excels, etc.

Model Analysis

- How to digitally express and analyze models?

Model Validation and Verification

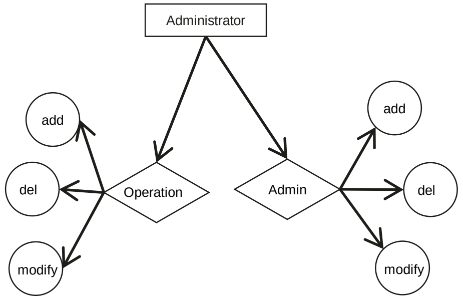
- Does the computational model reflect reality correctly?

Recorded in json, graph db, xml, code, etc.

3.1 Design for Dependability and the Models for Verification Purpose

Concepts and Requirements

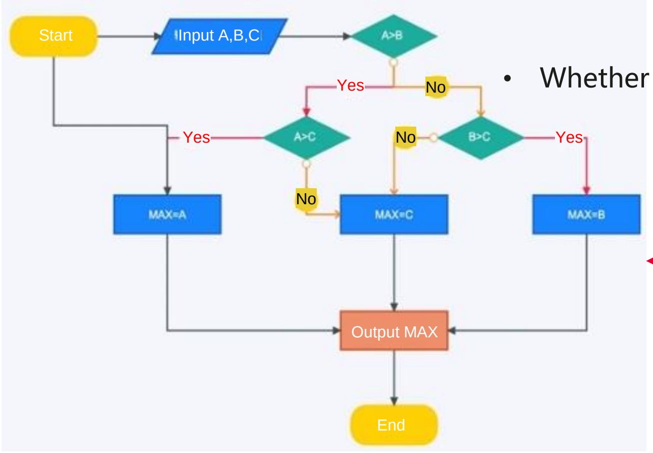
Formalized User requirement



- Whether the requirement is accurate?
- Whether the requirement is complete?

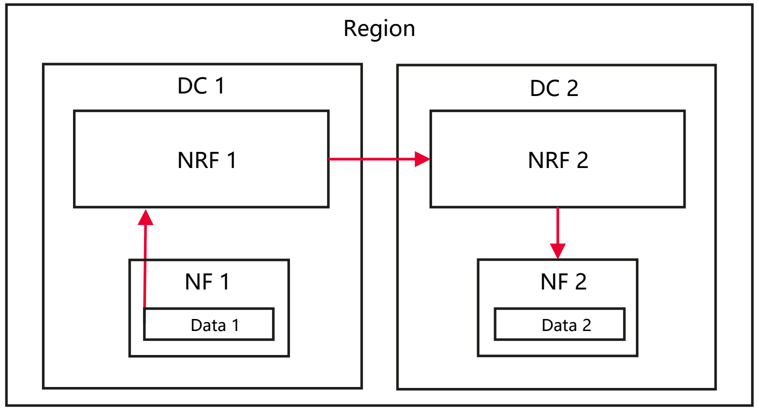
- **Whether the alg satisfies the requirement?**

System Workflow



- Whether the workflow/algorithm is correct?

System Arch

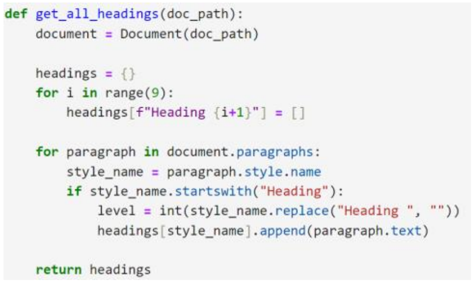


- Whether the arch is reliable?

- **Whether the arch satisfies the requirement?**

- **Whether the code matches the system arch and workflow?**

Code Implementation

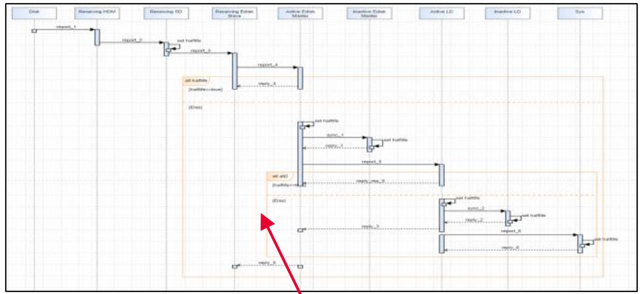


```python
def get_all_headings(doc_path):
    document = Document(doc_path)

    headings = {}
    for i in range(9):
        headings[f"Heading {i+1}"] = []

    for paragraph in document.paragraphs:
        style_name = paragraph.style.name
        if style_name.startswith("Heading"):
            level = int(style_name.replace("Heading ", ""))
            headings[style_name].append(paragraph.text)

    return headings
```

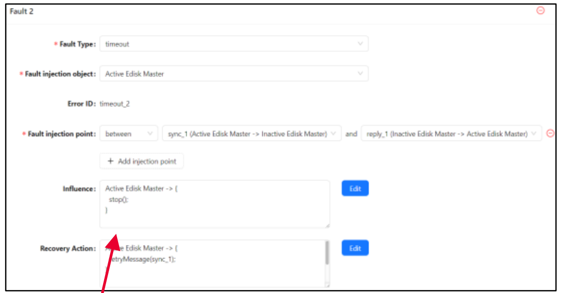
- Whether the code is bug free?

[Data storage] Discovery and verification of the scenario where messages are lost

Reliability modeling based on sequence diagrams and automatic generation of BIP code, accurately identifying the root cause of faults and modifying the design.



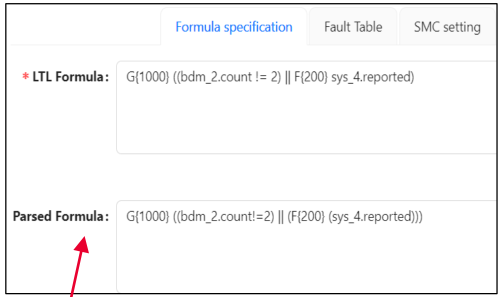
Building a Sequence Diagram Model



Fault 2

- * Fault Type: timeout
- * Fault injection object: Active Edisk Master
- Error ID: timeout_2
- * Fault injection point: between | sync_1 (Active Edisk Master -> Inactive Edisk Master) | and | reply_1 (Inactive Edisk Master -> Active Edisk Master)
- + Add injection point
- Influence: Active Edisk Master -> { stop(); } [Edit]
- Recovery Action: Active Edisk Master -> { retryMessage(sync_1); } [Edit]

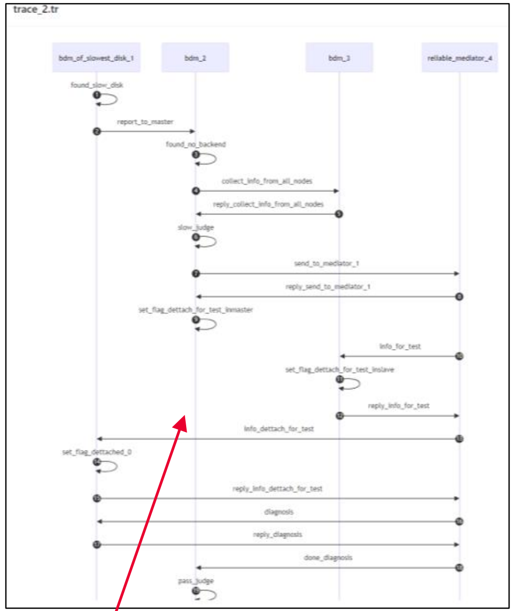
Fault Definition



Formula specification | Fault Table | SMC setting

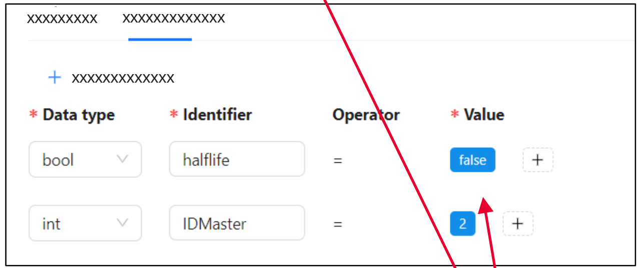
- * LTL Formula: G[1000] ((bdm_2.count != 2) || F[200] sys_4.reported)
- Parsed Formula: G[1000] ((bdm_2.count!=2) || (F[200] (sys_4.reported)))

Specification



trace_2.tr

Faulty path display

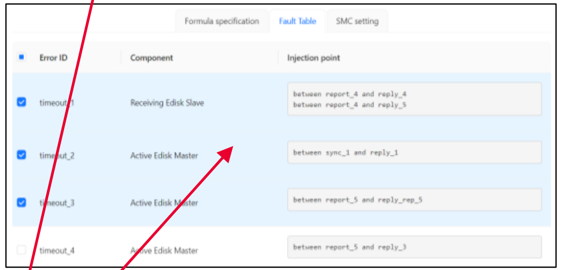


xxxxxxxxx | xxxxxxxxxxxxx

+ xxxxxxxxxxxxx

* Data type	* Identifier	Operator	* Value
bool	halflife	=	false
int	IDMaster	=	2

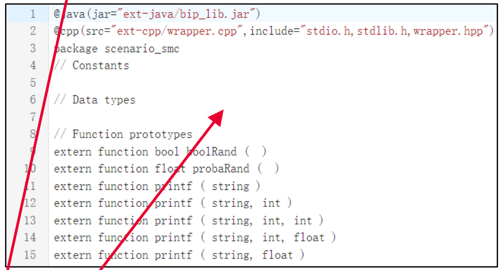
Status parameter definition



Formula specification | Fault Table | SMC setting

	Error ID	Component	Injection point
☑	timeout_1	Receiving Edisk Slave	between report_4 and reply_4 between report_4 and reply_5
☑	timeout_2	Active Edisk Master	between sync_1 and reply_1
☑	timeout_3	Active Edisk Master	between report_5 and reply_rep_5
☐	timeout_4	Active Edisk Master	between report_5 and reply_3

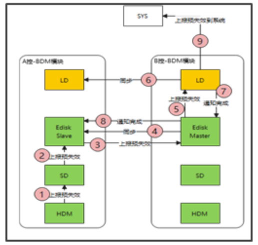
Fault injection selection



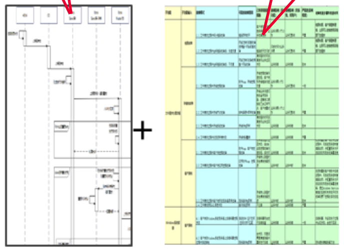
```
@java(jar="ext-java/bip_lib.jar")
@cpp(src="ext-cpp/wrapper.cpp",include="stdio.h,stdlib.h,wrapper.hpp")
package scenario_smc
// Constants

// Data types

// Function prototypes
extern function bool boolRand (  )
extern function float probaRand (  )
extern function printf ( string )
extern function printf ( string, int )
extern function printf ( string, int, int )
extern function printf ( string, int, float )
extern function printf ( string, float )
```

Automated code generation

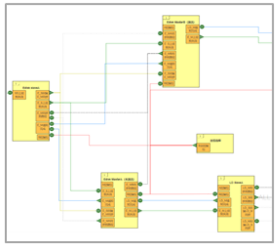
The tools and platform



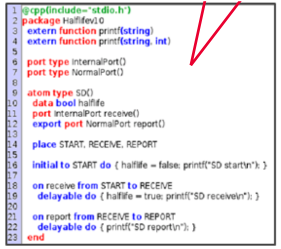
System Arch



SD + Table

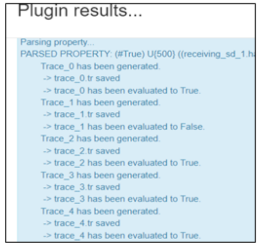


BIP Diagram



```
@cpp(include="stdio.h")
package Halflifev10
  extern function printf(string)
  extern function printf(string, int)

  port type InternalPort()
  port type NormalPort()

  atom type SD()
    data bool halflife
    port InternalPort receive()
    export port NormalPort report()

    place START, RECEIVE, REPORT

    initial to START do { halflife = false; printf("SD start\n"); }

    on receive from START to RECEIVE
      delayable do { halflife = true; printf("SD receive\n"); }

    on report from RECEIVE to REPORT
      delayable do { printf("SD report\n"); }
  end
```

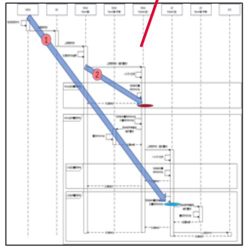
BIP Code



Plugin results...

```
Parsing property...
PARSED PROPERTY: (#True) U{500} ((receiving_sd_1.h
        Trace_0 has been generated.
         -> trace_0.tr saved
         -> trace_0 has been evaluated to True.
        Trace_1 has been generated.
         -> trace_1.tr saved
         -> trace_1 has been evaluated to False.
        Trace_2 has been generated.
         -> trace_2.tr saved
         -> trace_2 has been evaluated to True.
        Trace_3 has been generated.
         -> trace_3.tr saved
         -> trace_3 has been evaluated to True.
        Trace_4 has been generated.
         -> trace_4.tr saved
         -> trace_4 has been evaluated to True.
```

BIP Verification

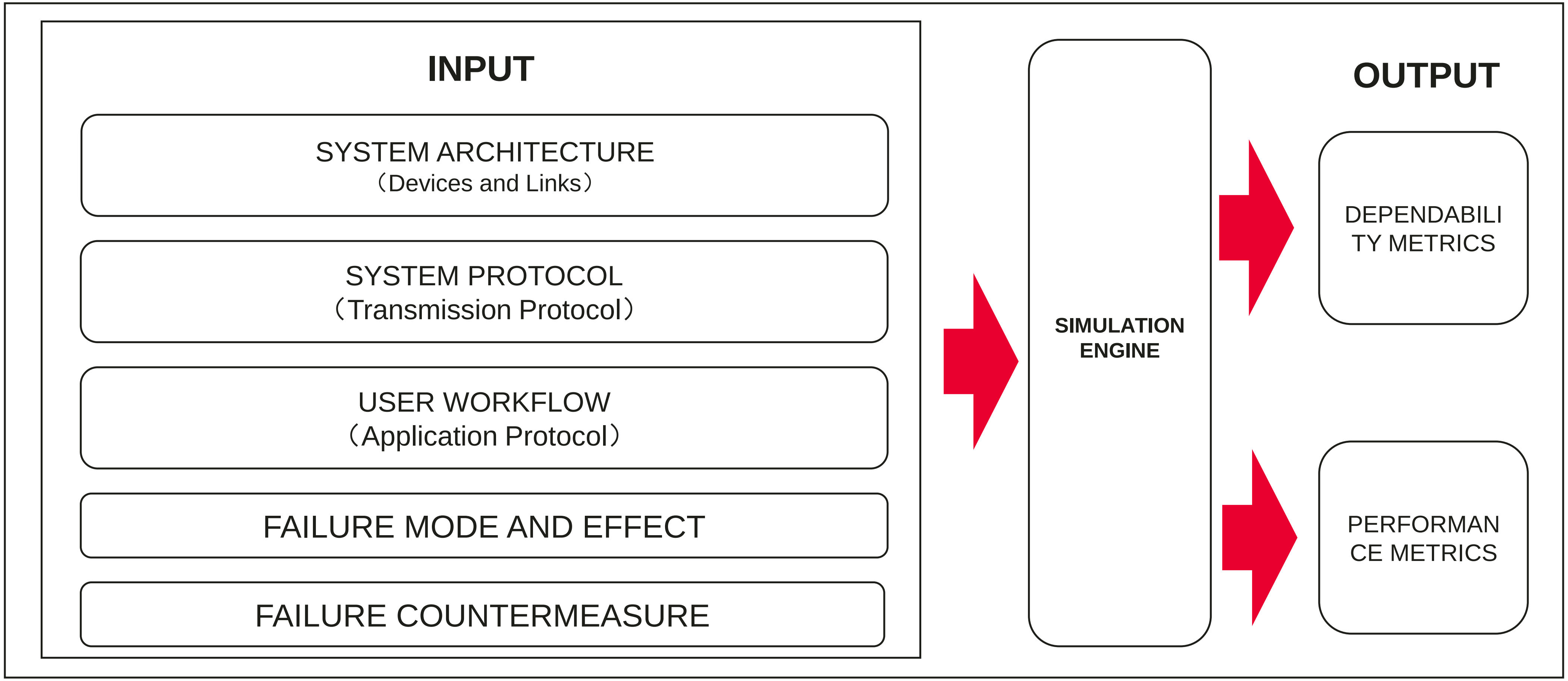


Fault Path

Revise design

Revise design

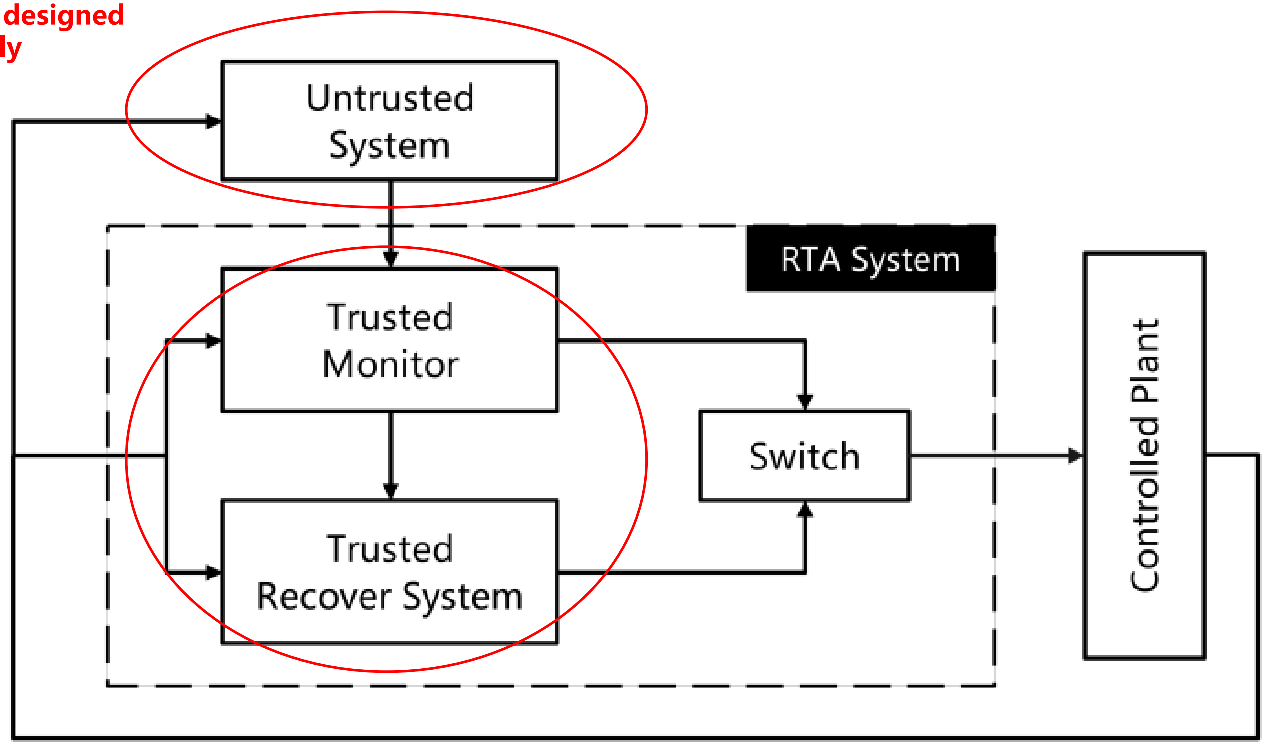
3.2 Design for Dependability and the Models for Validation Purpose



3.3 Design for Dependability and the Runtime Assurance System (for ADN)

- The network management system could be a E2E system without rules: no carefully designed KPI analysis algorithms and no carefully designed fault localization algorithms.

- Rule-based system step back to guard the red line of the system and make sure no worst case happen.



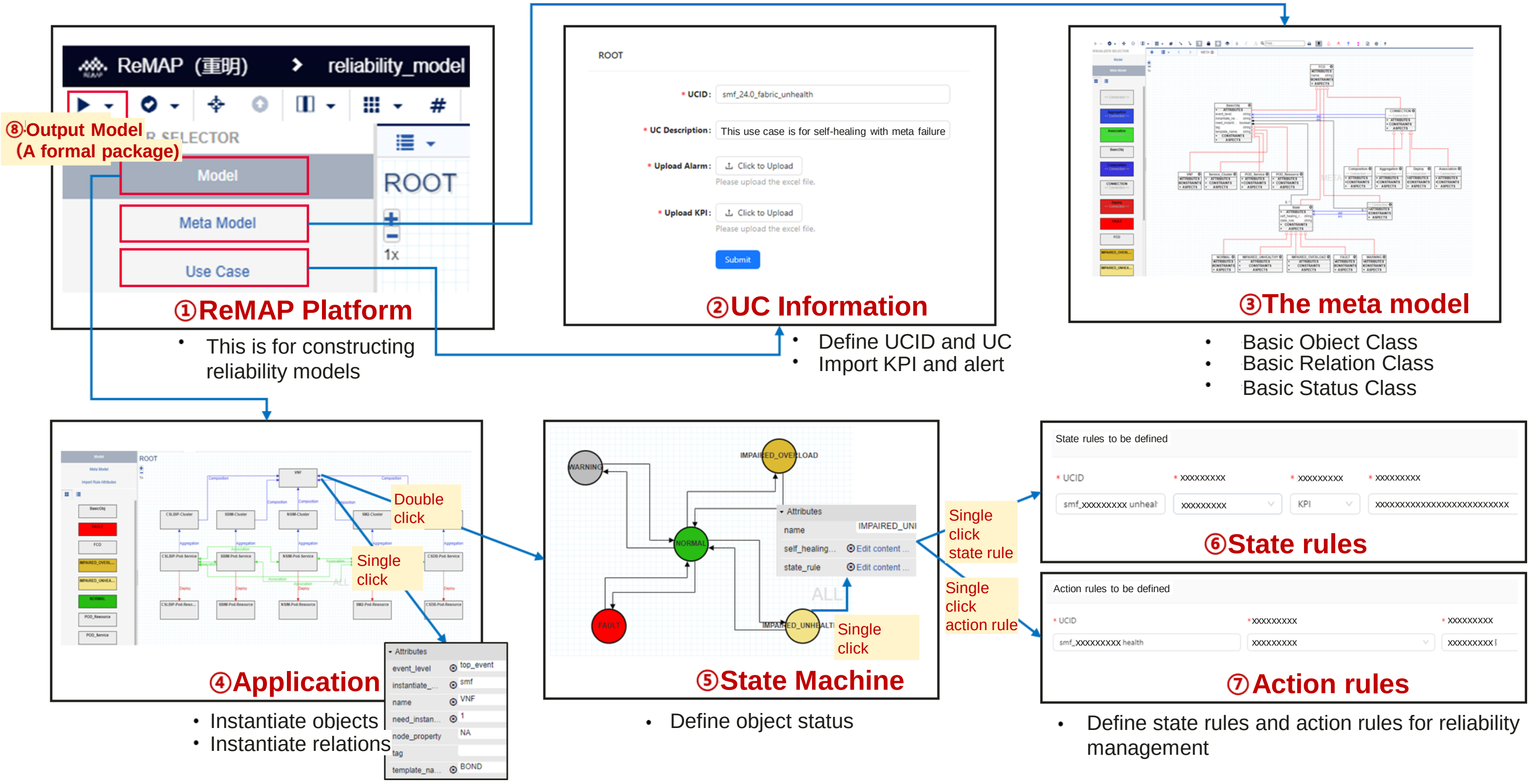
This architecture is still controversial but already meet the requirements from different stakeholders:

- System engineers would like to accomplish a function in the simplest way – the untrusted system be a E2E function;
- Reliability engineers would like to make sure no bad thing happen – the RTA system be the safe guard;

[Cloud Core] Object status relationship modeling, supporting complex fault diagnosis and self-healing

□ **Based on the ReMAP, define the normal/abnormal status and judgment conditions of system objects, define the fault propagation relationship between objects, generate the reliability model adaptation package, and instantiate the model based on the live network topology.**

1. Design time: ReMAP modeling generates reliability models (adaptation package).
2. Run time: Import the mediation package to the simulation system.
3. Run time: model-based real-time fault detection/diagnosis/self-healing simulation.



①ReMAP Platform

- This is for constructing reliability models

ROOT

* UCID: smf_24.0_fabric_unhealth

* UC Description: This use case is for self-healing with meta failure

* Upload Alarm: Click to Upload
Please upload the excel file.

* Upload KPI: Click to Upload
Please upload the excel file.

Submit

②UC Information

- Define UCID and UC
- Import KPI and alert

③The meta model

- Basic Object Class
- Basic Relation Class
- Basic Status Class

Attributes

event_level	top_event
instantiate_...	smf
name	VNF
need_instan...	1
node_property	NA
tag	
template_na...	BOND

④Application

- Instantiate objects
- Instantiate relations

Attributes

name	IMPAIRED_UNI
self_healing...	Edit content ...
state_rule	Edit content ...

⑤State Machine

- Define object status

State rules to be defined

* UCID	* xxxxxxxxx	* xxxxxxxxx	* xxxxxxxxx
smf_xxxxxxxxx unheal	xxxxxxxxx	KPI	xxxxxxxxxxxxxxxxxxxxxxxxxx

⑥State rules

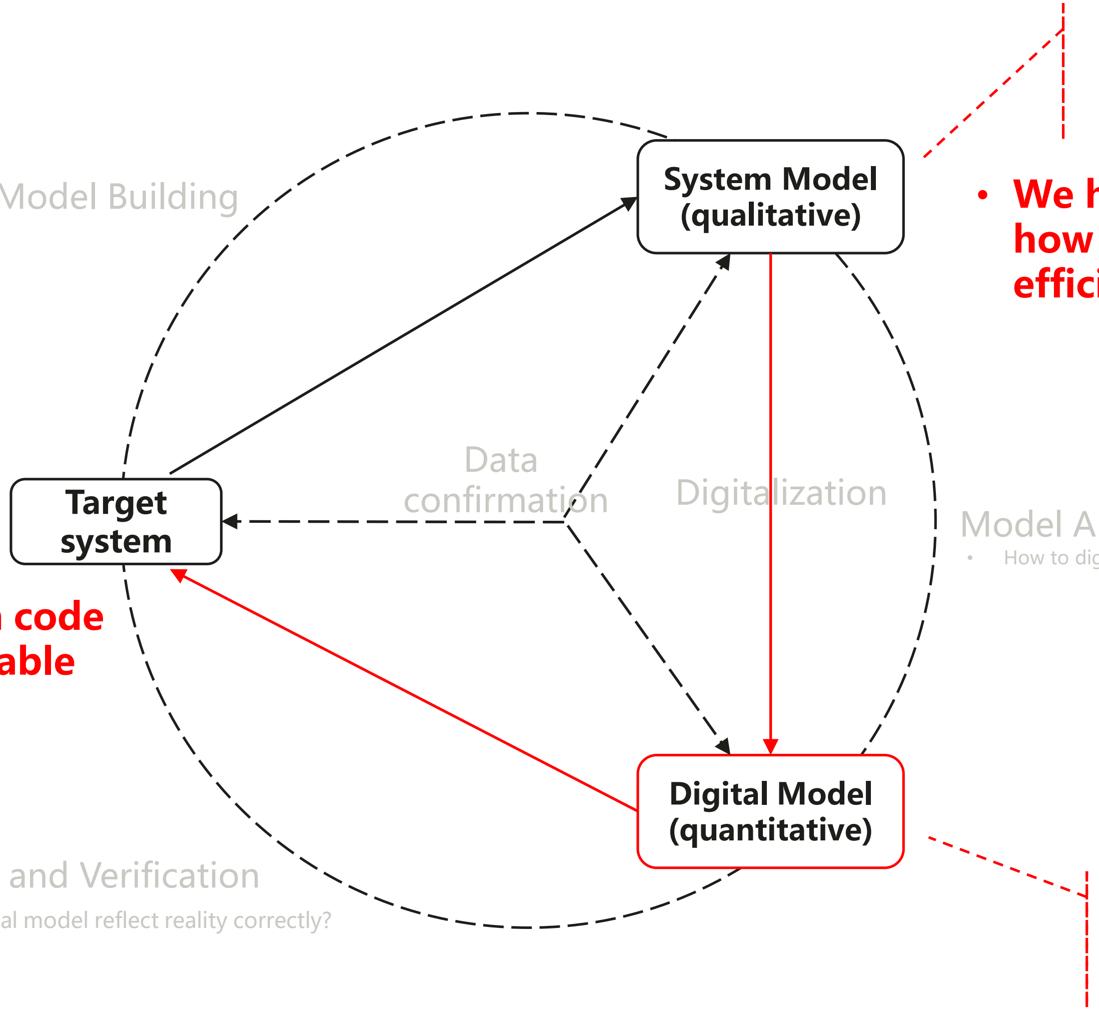
Action rules to be defined

* UCID	* xxxxxxxxx	* xxxxxxxxx
smf_xxxxxxxxx health	xxxxxxxxx	xxxxxxxxx

⑦Action rules

- Define state rules and action rules for reliability management

4.1 Challenges: Design for Dependability the Models



Recorded in docs, ppts, excels, etc.

- **We have countless existing system documents, how to use them to build models more efficiently and benefit future system design?**

Model Analysis

- How to digitally express and analyze models?

- **How to verify that the system code match the design with acceptable effort ?**

Model Validation and Verification

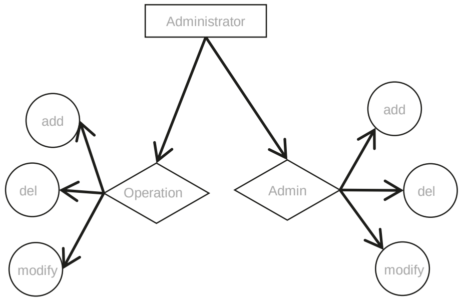
- Does the computational model reflect reality correctly?

Recorded in json, graph db, xml, code, etc.

4.2 Challenges: Design for Dependability the Models for Verification Purpose

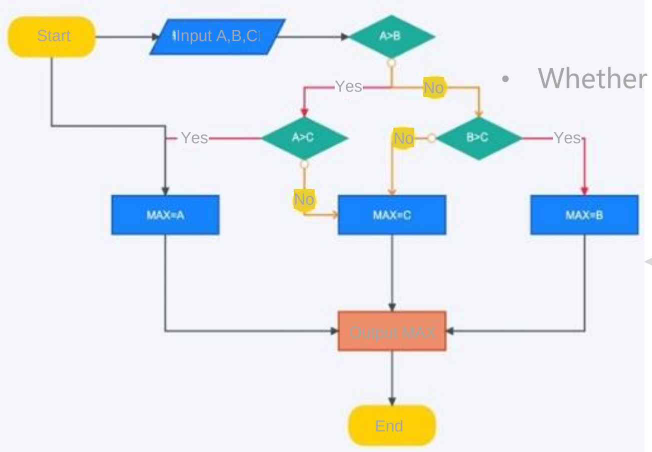
Concepts and Requirements

Formalized User requirement



- Whether the requirement is accurate?
- Whether the requirement is complete?

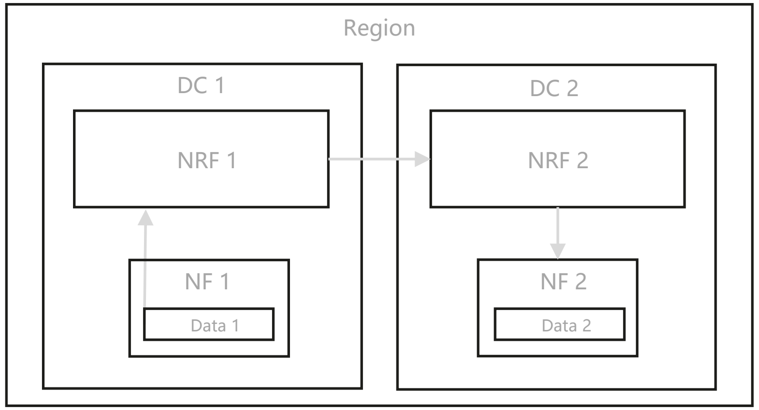
System Workflow



- Whether the workflow/algorithm is correct?

- **Whether the alg satisfies the requirement?**

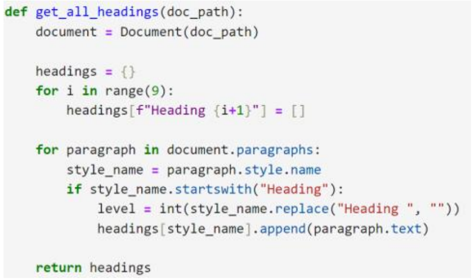
System Arch



- Whether the arch is reliable?

- **Whether the arch satisfies the requirement?**

Code Implementation



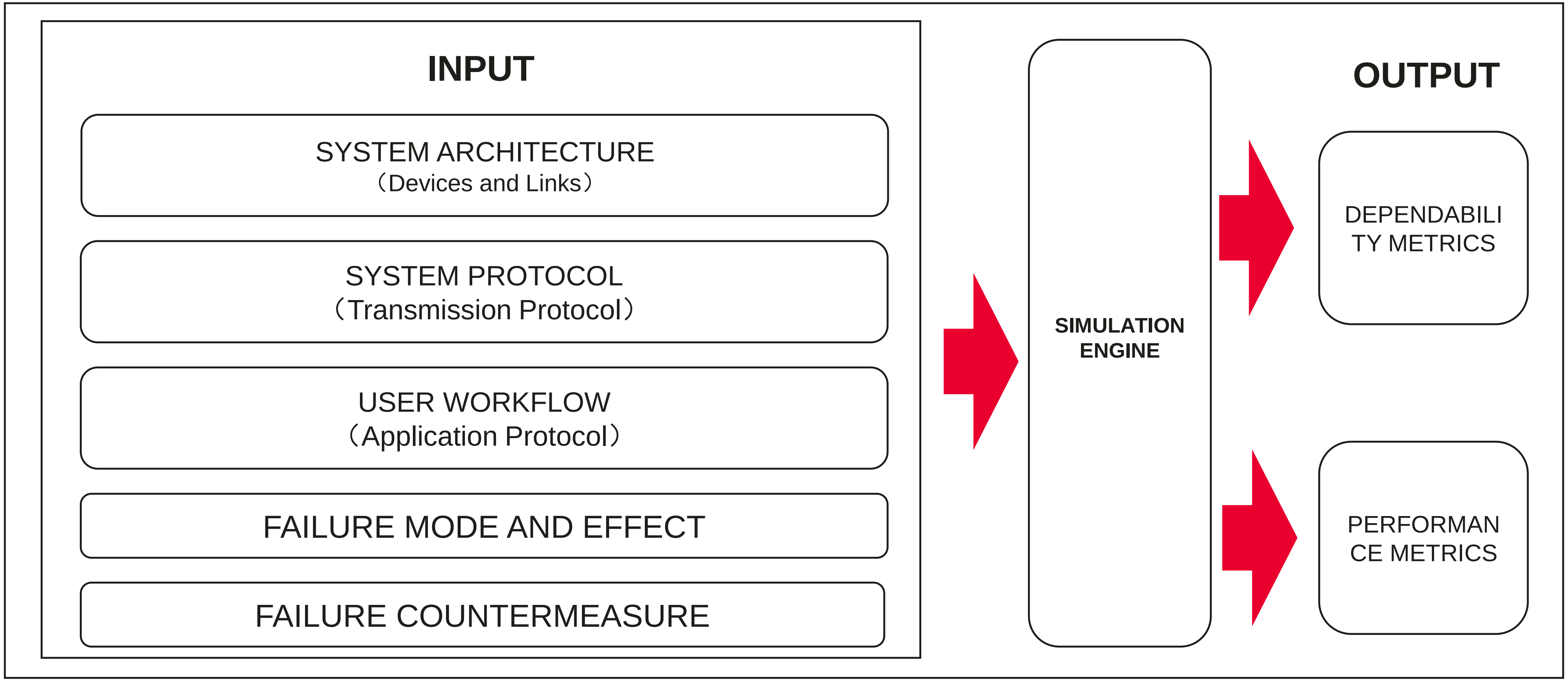
```python
def get_all_headings(doc_path):
    document = Document(doc_path)

    headings = {}
    for i in range(9):
        headings[f"Heading {i+1}"] = []

    for paragraph in document.paragraphs:
        style_name = paragraph.style.name
        if style_name.startswith("Heading"):
            level = int(style_name.replace("Heading ", ""))
            headings[style_name].append(paragraph.text)

    return headings
```

- Whether the code is bug free?

- **Whether the code matches the system arch and workflow?**

- **Formal Modeling is not easily acceptable by engineers. It is currently used for designing critical system or modules. Could there be a way to reduce the cost and make it acceptable by engineers. (Building models is time-consuming and a reliable model does not necessarily lead to a reliable system. The whole idea could be wrong: verification is not validation)**

4.3 Challenges: Design for Dependability the Models for Validation Purpose

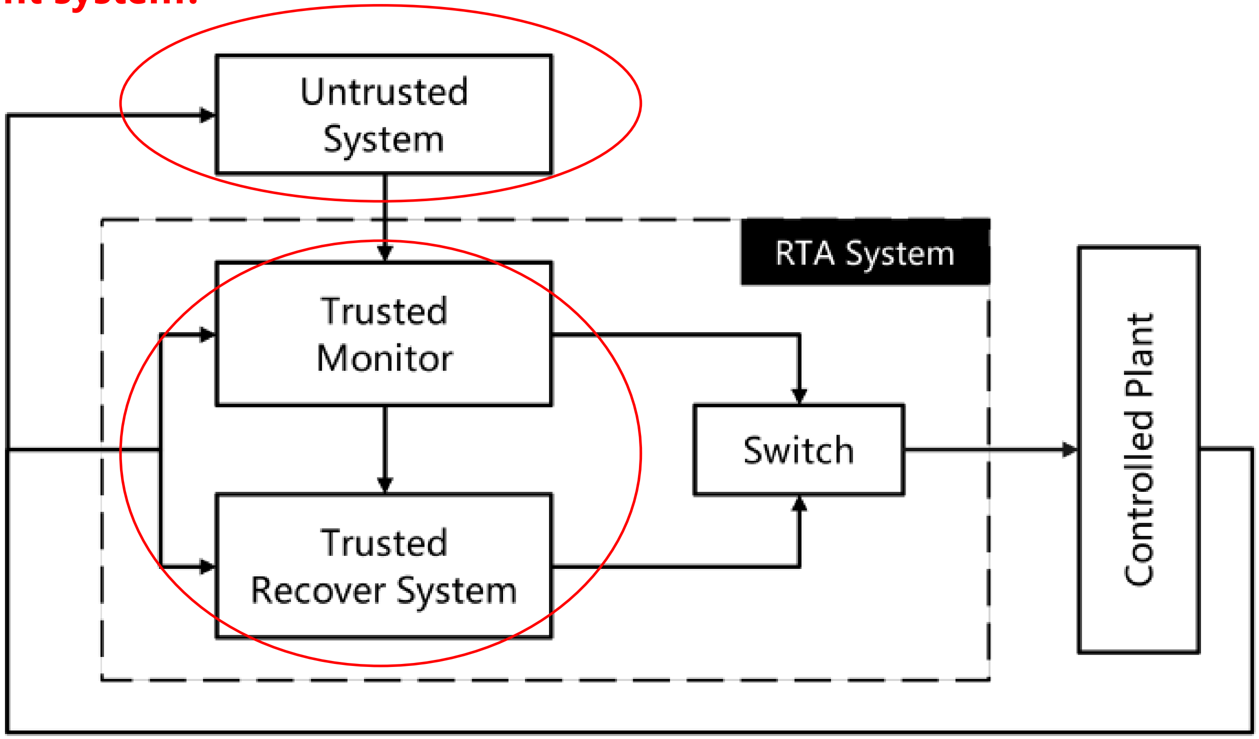


- **We require a unified simulation system that would enable the simulation of different systems. For the arch and protocol we could use meta-model (objects, relations, etc.). But for user workflow simulation, techniques are required in the presence of too many user workflows in order to reduce manual effort.**

4.4 Challenge：Design for Dependability the Runtime Assurance System

- How to collect the data for training the E2E network management system?
- How to make it generalize?

- How to make sure the action is benign? Or detect the harmful action.



- The challenges all lead to a simulation system that nicely mimic certain aspects of the original system. How to balance the cost and accuracy of building a simulation system.

Thank you.

Bring digital to every person, home and organization for a fully connected, intelligent world.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

HUAWEI