A TRUST-BY-DESIGN FRAMEWORK FOR THE INTERNET OF THINGS

ESR3 - Davide Ferraris

PhD Student @ University of Malaga, NICS lab

Tutors: Prof. Javier Lopez , Dr. Carmen Fernandez Gago

Contents

- Introduction
  - Trust
  - Internet of Things
- Architecture Framework
  - K Model
  - Transversal Activities
- Use Case Scenario
- Conclusion

Introduction

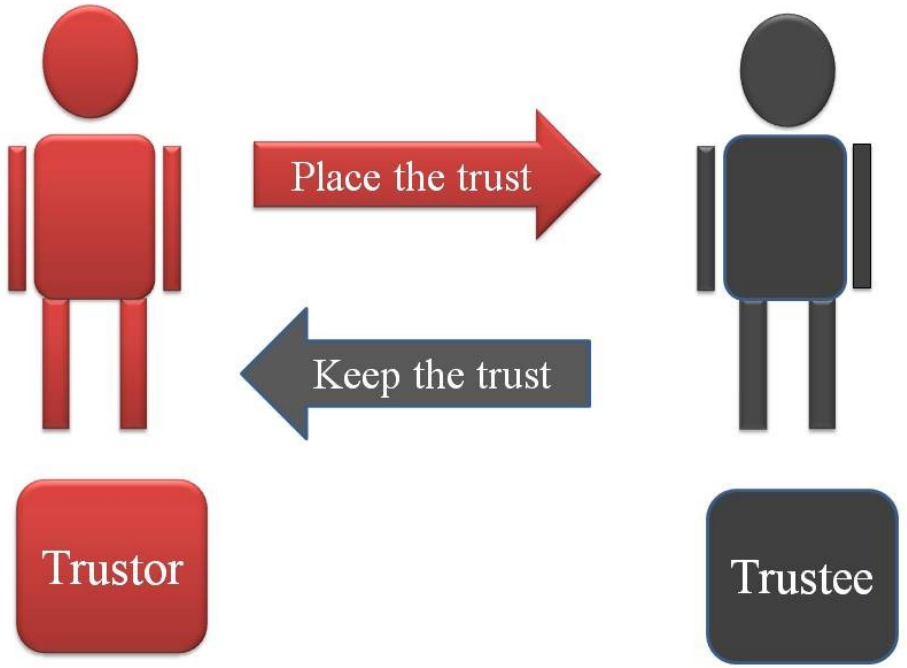
- Introduction
  - Trust
  - Internet of Things
- Architecture Framework
  - K Model
  - Transversal Activities
- Use Case Scenario
- Conclusion

Trust

Trust is difficult to define because:

- “To believe that someone is good and honest and will not harm you, or that something is safe and reliable”.

- Trustor and Trustee



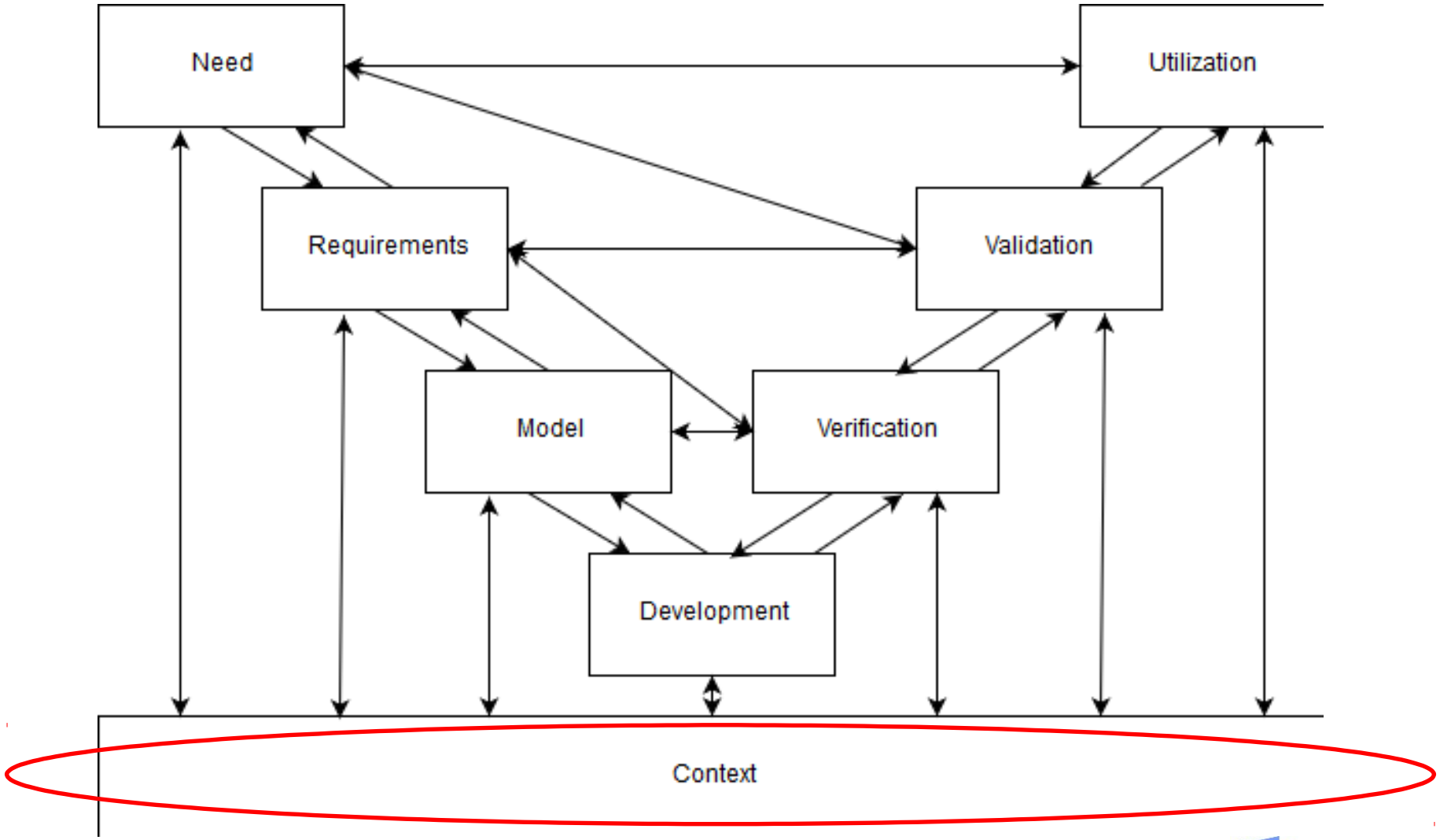
Internet of Things

- 20.4 billions of devices will be connected by 2020 (https://www.gartner.com/newsroom/id/3598917)
- Heterogeneity
- Dinamicity
- Communication
- Trust is **needed**

Architecture Framework

- Introduction
  - Trust
  - Internet of Things
- Architecture Framework
  - K Model
  - Transversal Activities
- Use Case Scenario
- Conclusion

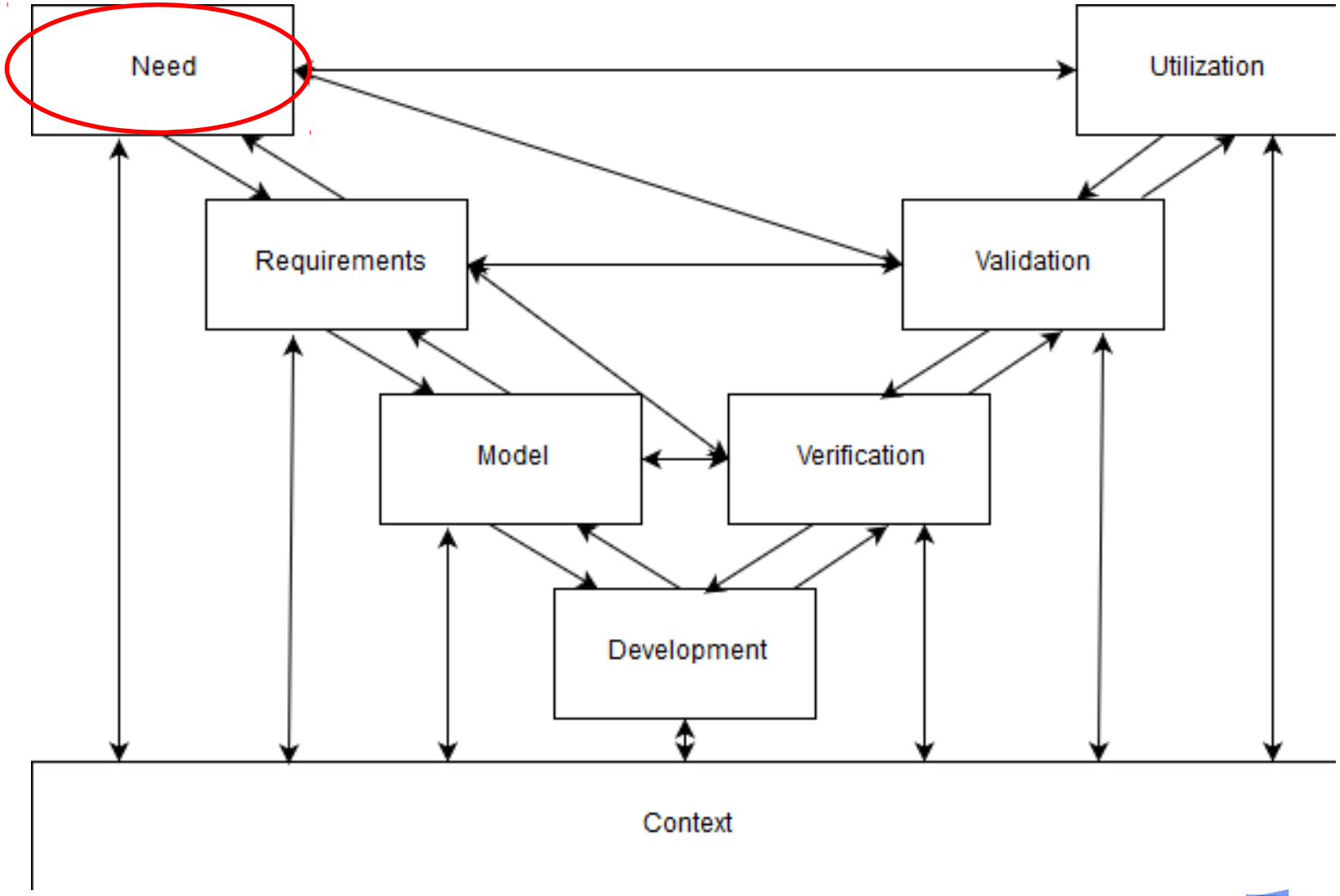
K Model


Need
Utilization
Requirements
Validation
Model
Verification
Development
Context

Context

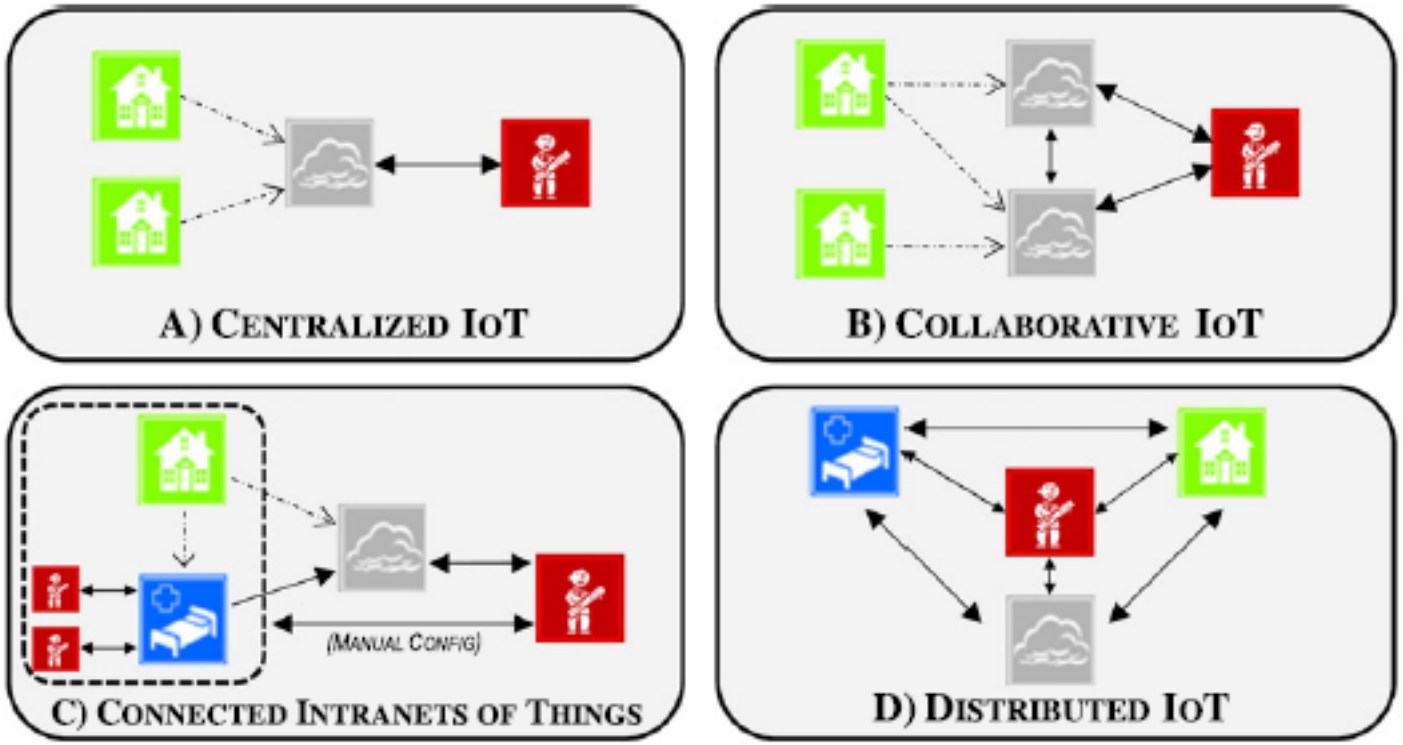
- **Always** present
- Environment
- Services
- Properties (alone or composition)
- Dynamic

K Model

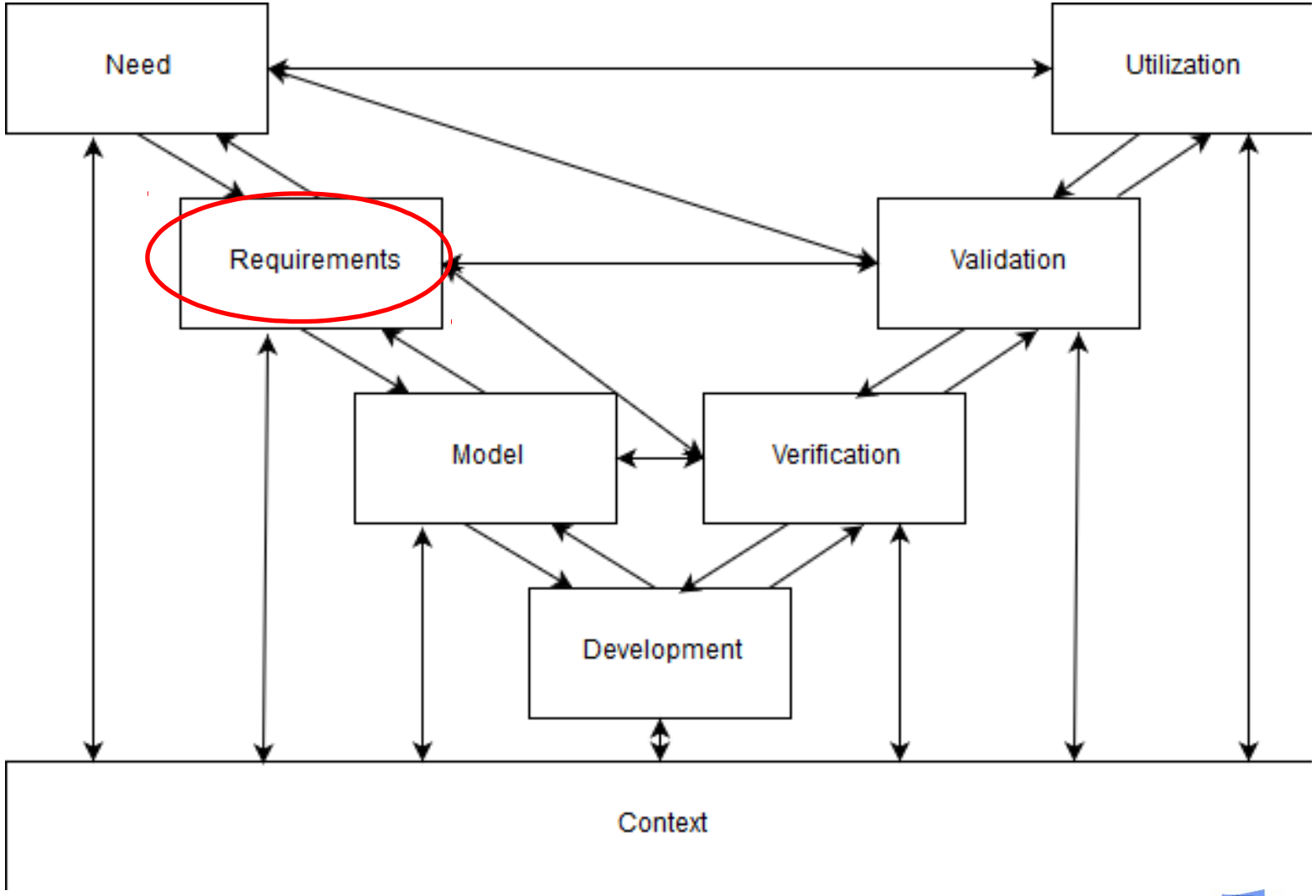

Need
Utilization
Requirements
Validation
Model
Verification
Development
Context

Need

- Characteristics of trust
- Type of Architecture
- Protocols



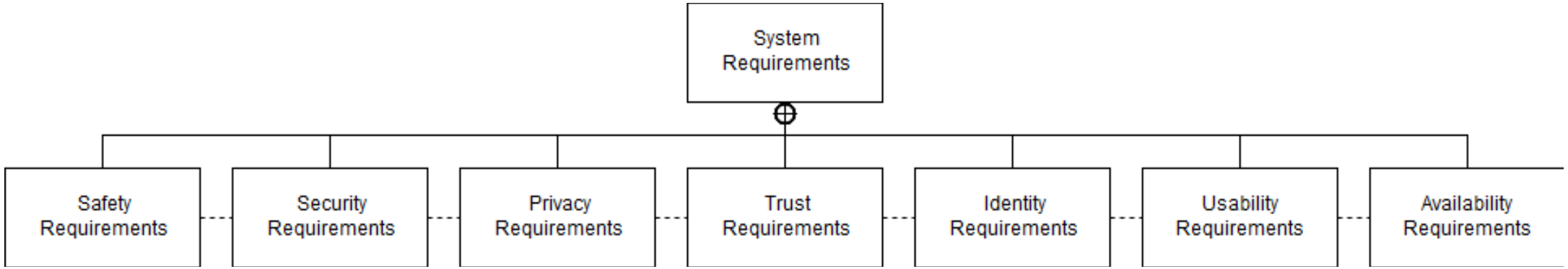
K Model


Need
Utilization
Requirements
Validation
Model
Verification
Development
Context

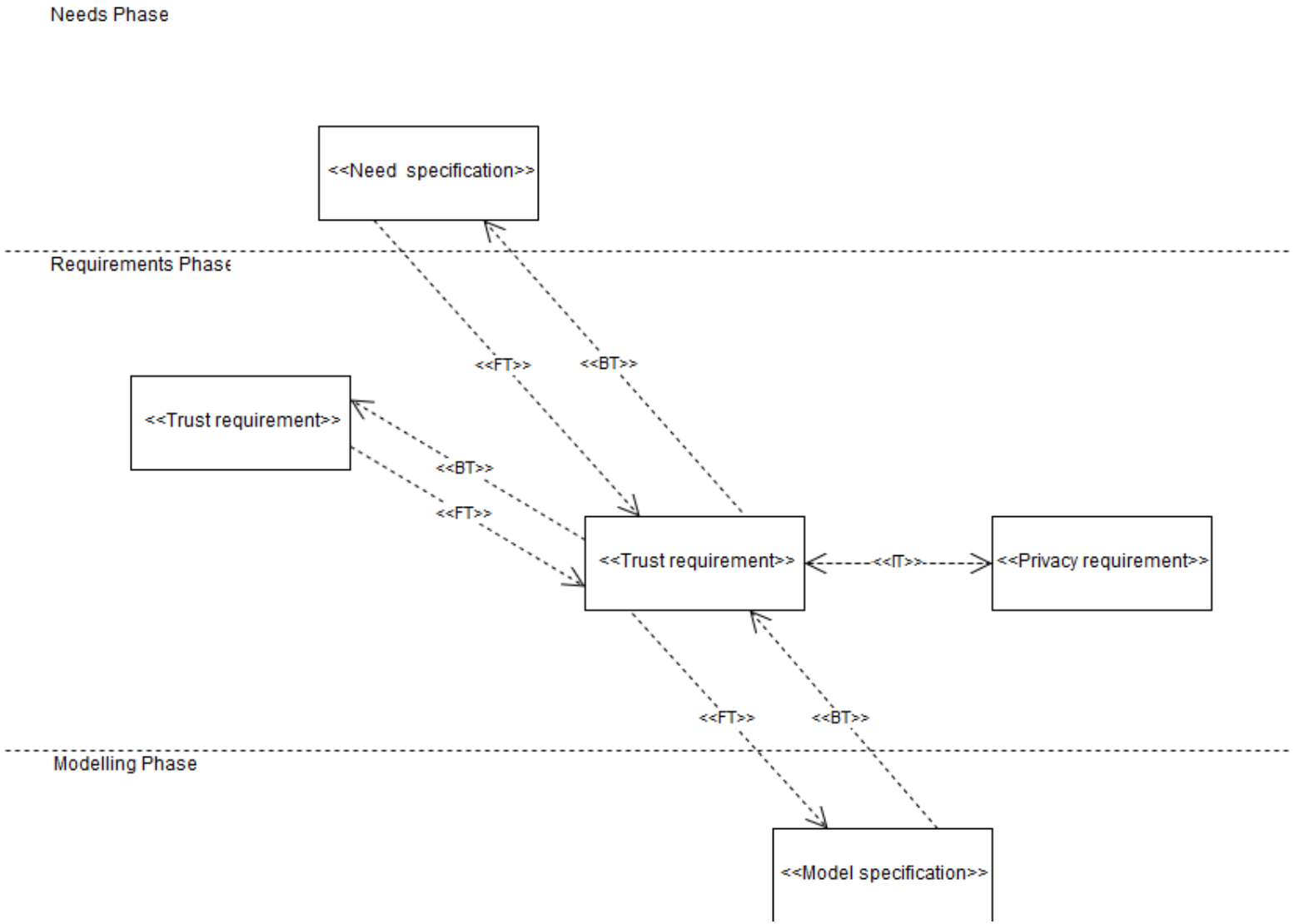
Requirements

- IEEE 830-1993 specification

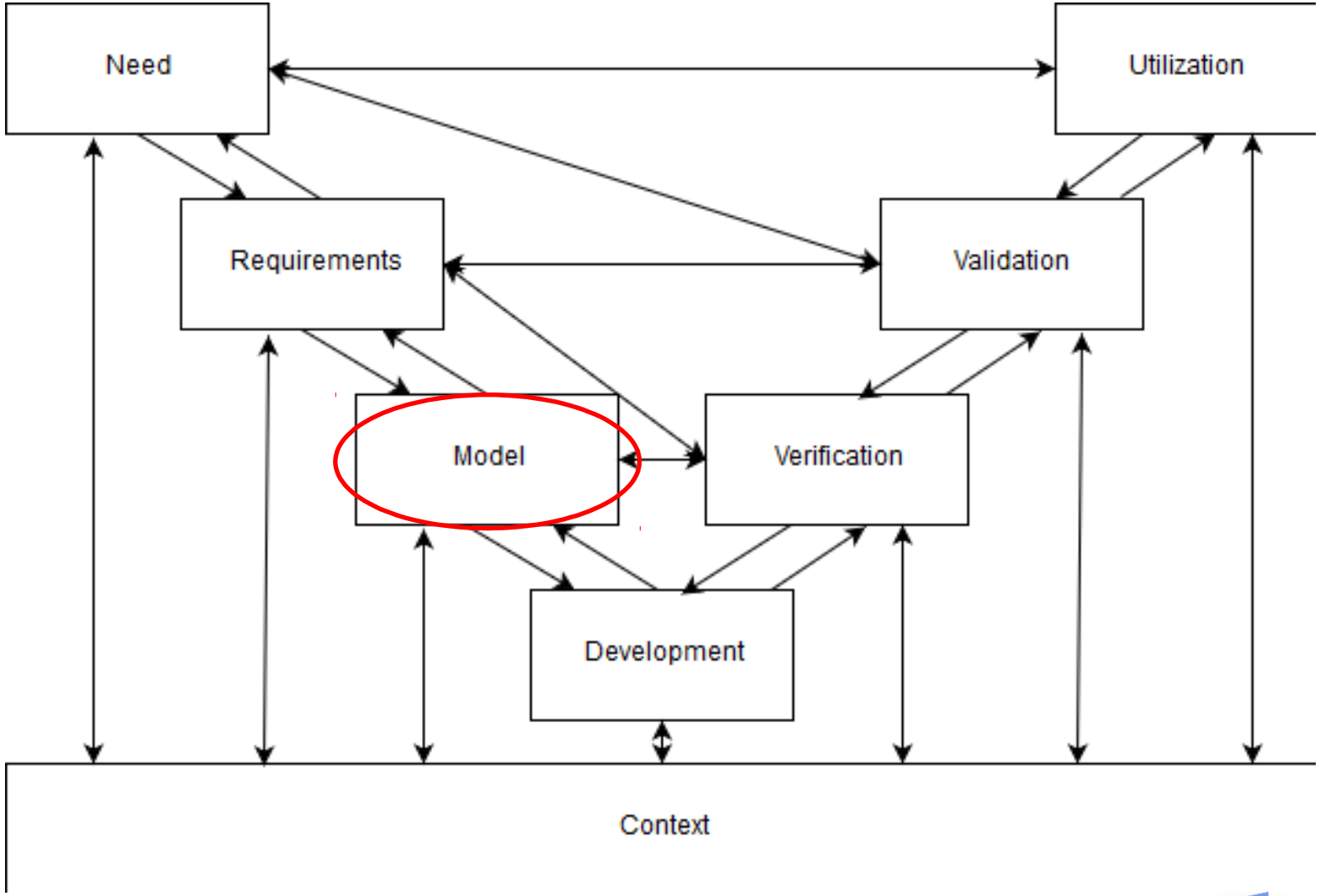
ID	Requirement specification	BT	FT	IT



Requirement Specification

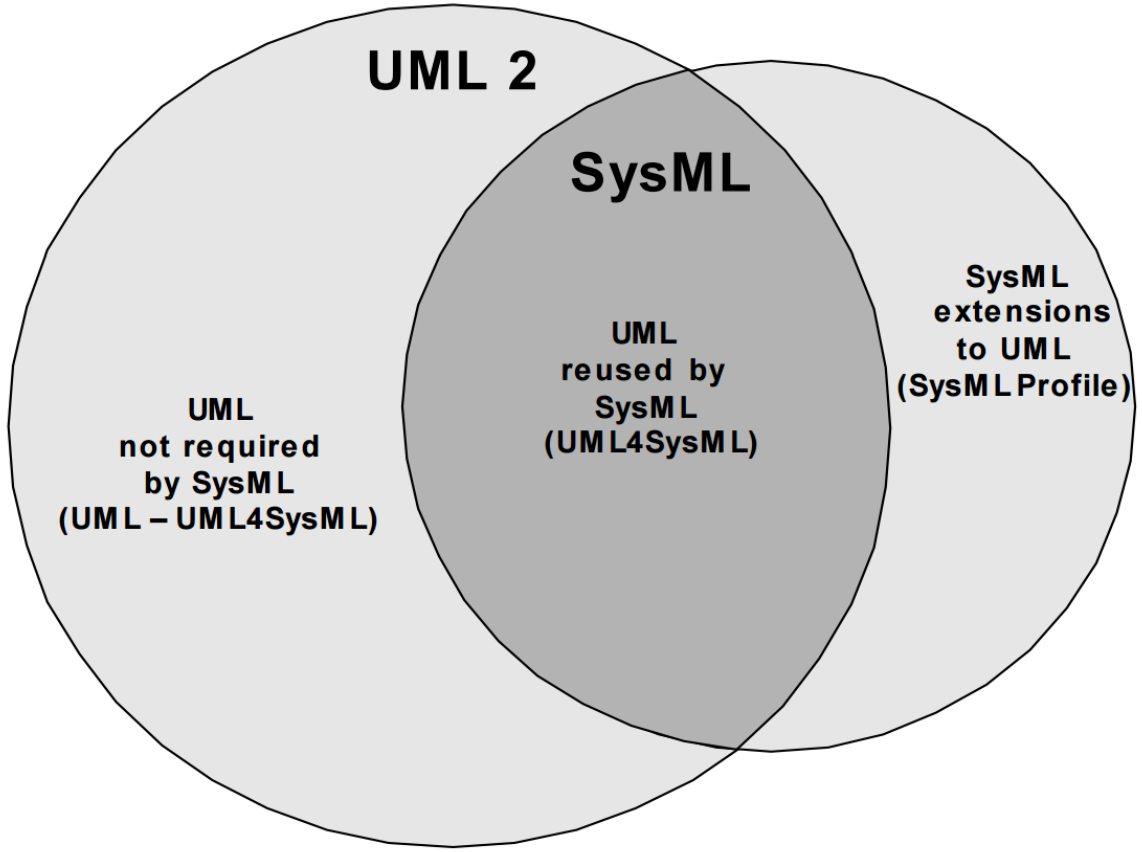


K Model

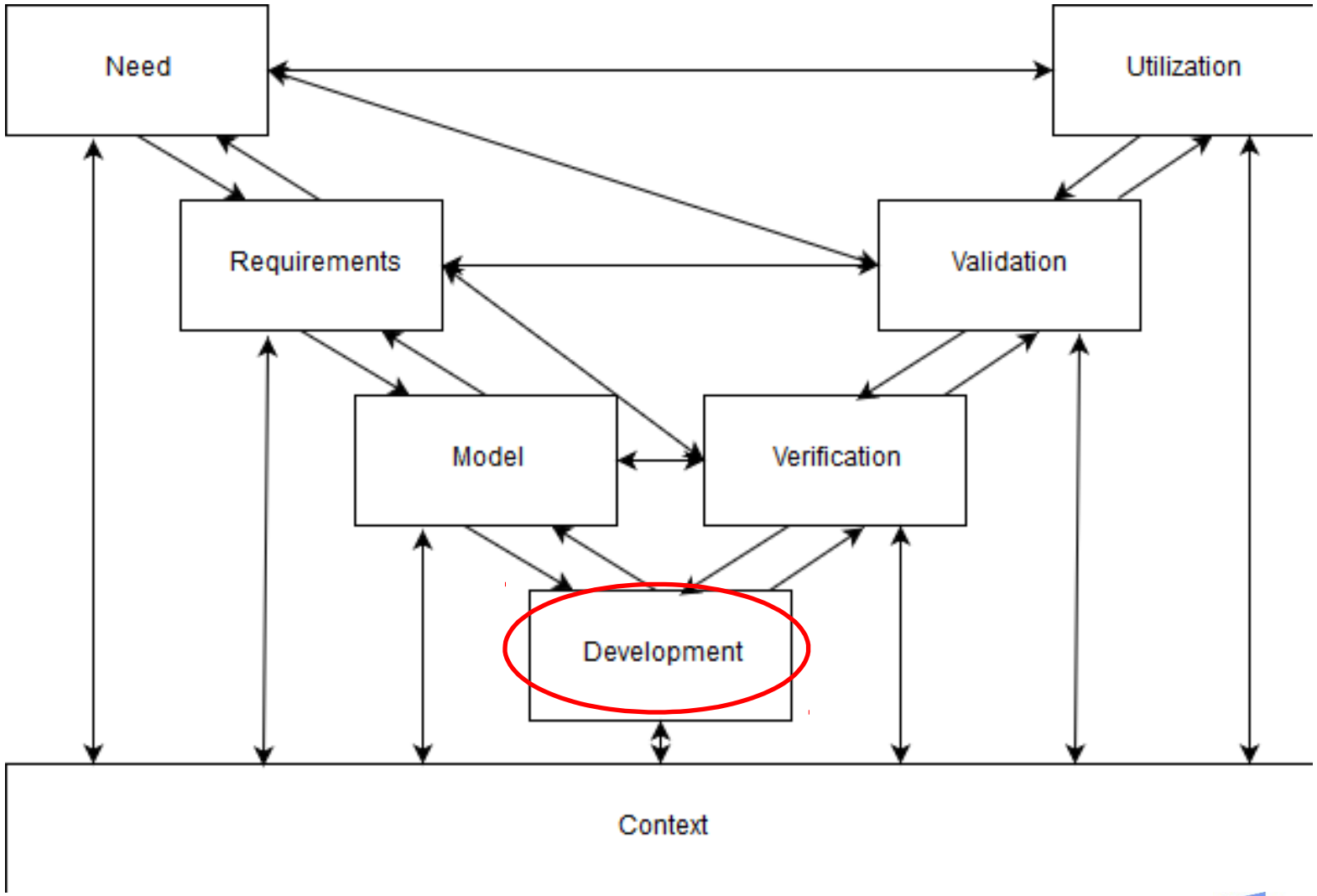

Need
Utilization
Requirements
Validation
Model
Verification
Development
Context

Model

- SysML
- Trust Models
  - Evaluation
  - Decision

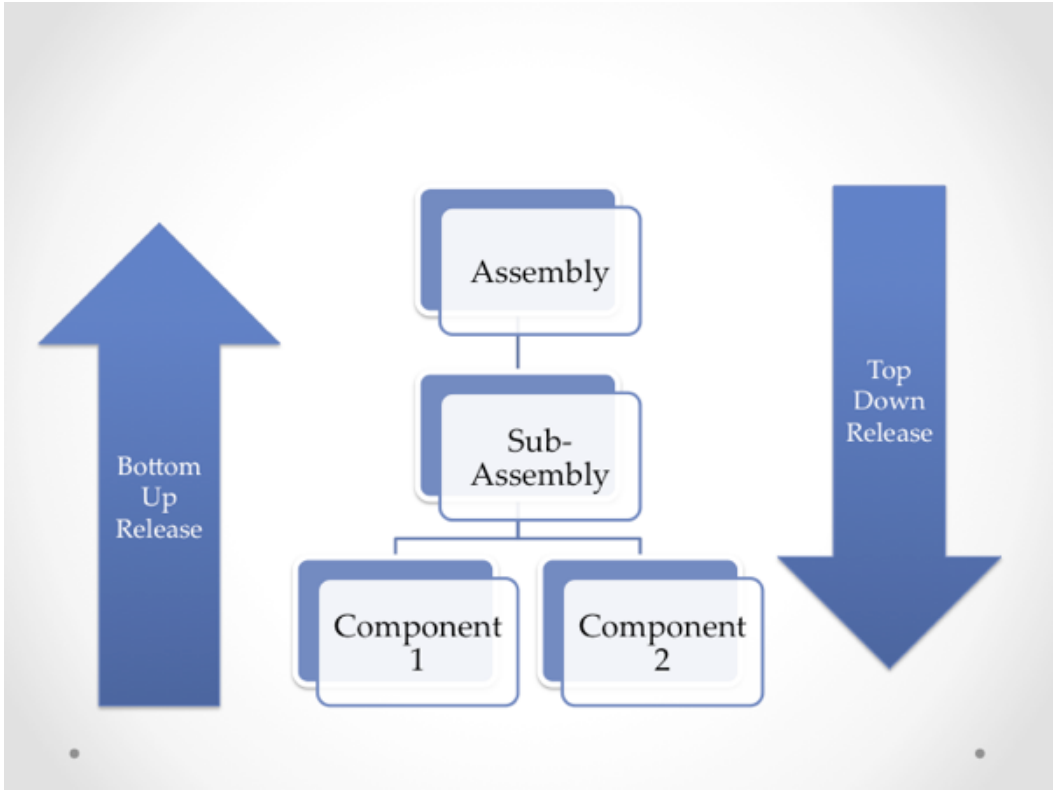


K Model

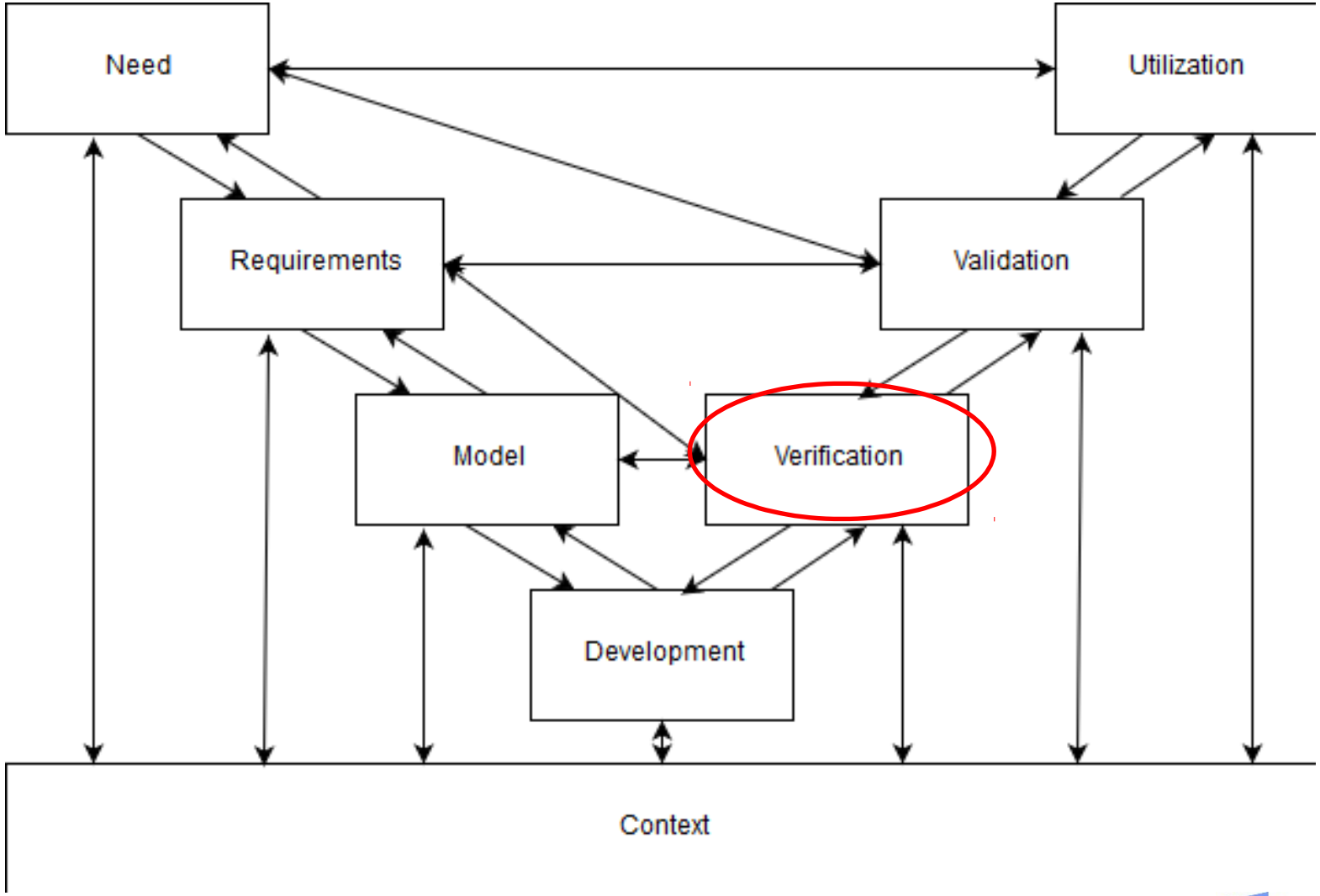

Need
Utilization
Requirements
Validation
Model
Verification
Development
Context

Development

- Top Down approach
- Bottom Up approach
- Depending on the previous and following phases
- Core of the framework
- Developer centric approach



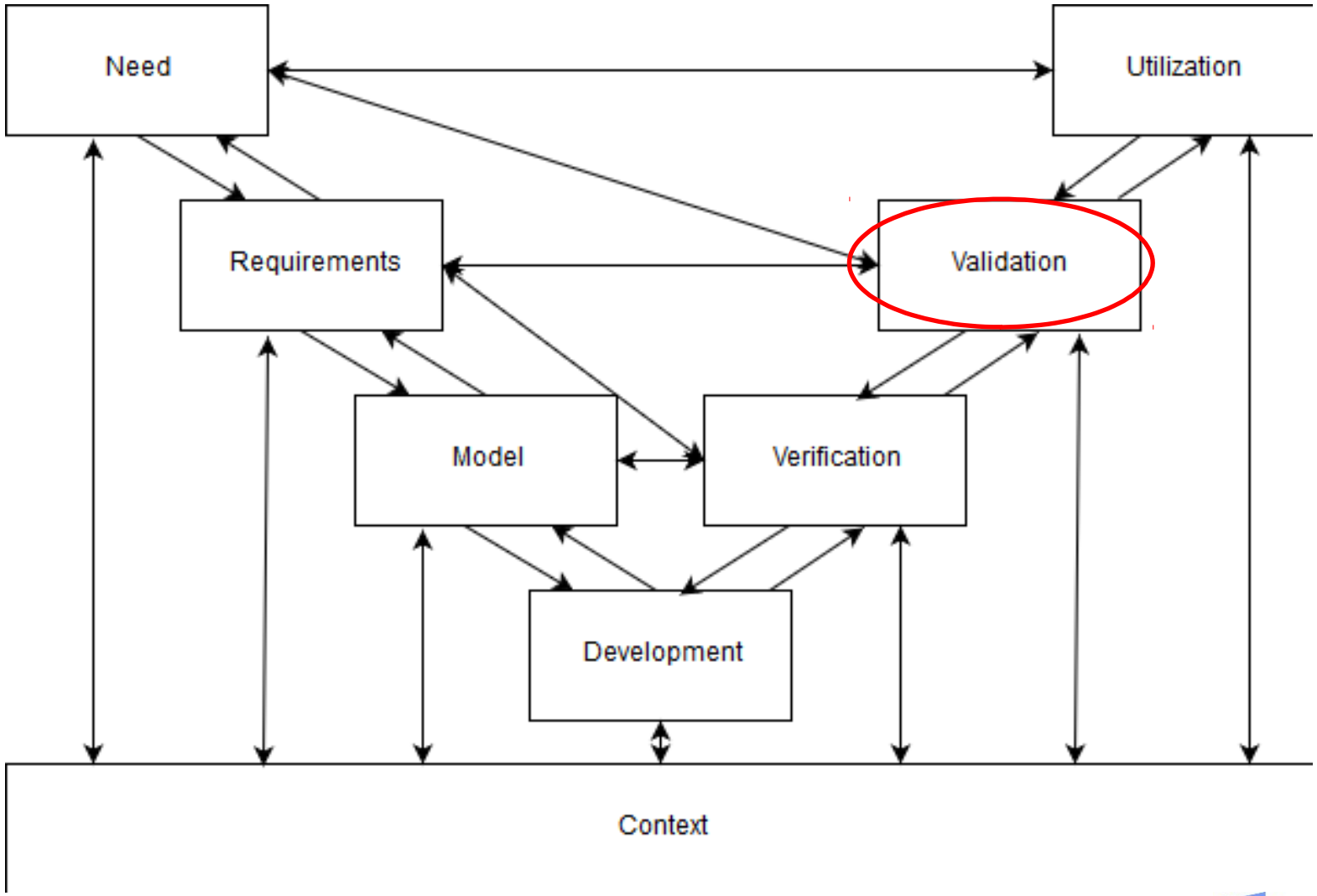
K Model


Need
Utilization
Requirements
Validation
Model
Verification
Development
Context

Verification

- Check if “the entity has been built right”
- Verification of the functionalities
- Verification of the requirements related to the system
- Developer point of view
- Intermediate product

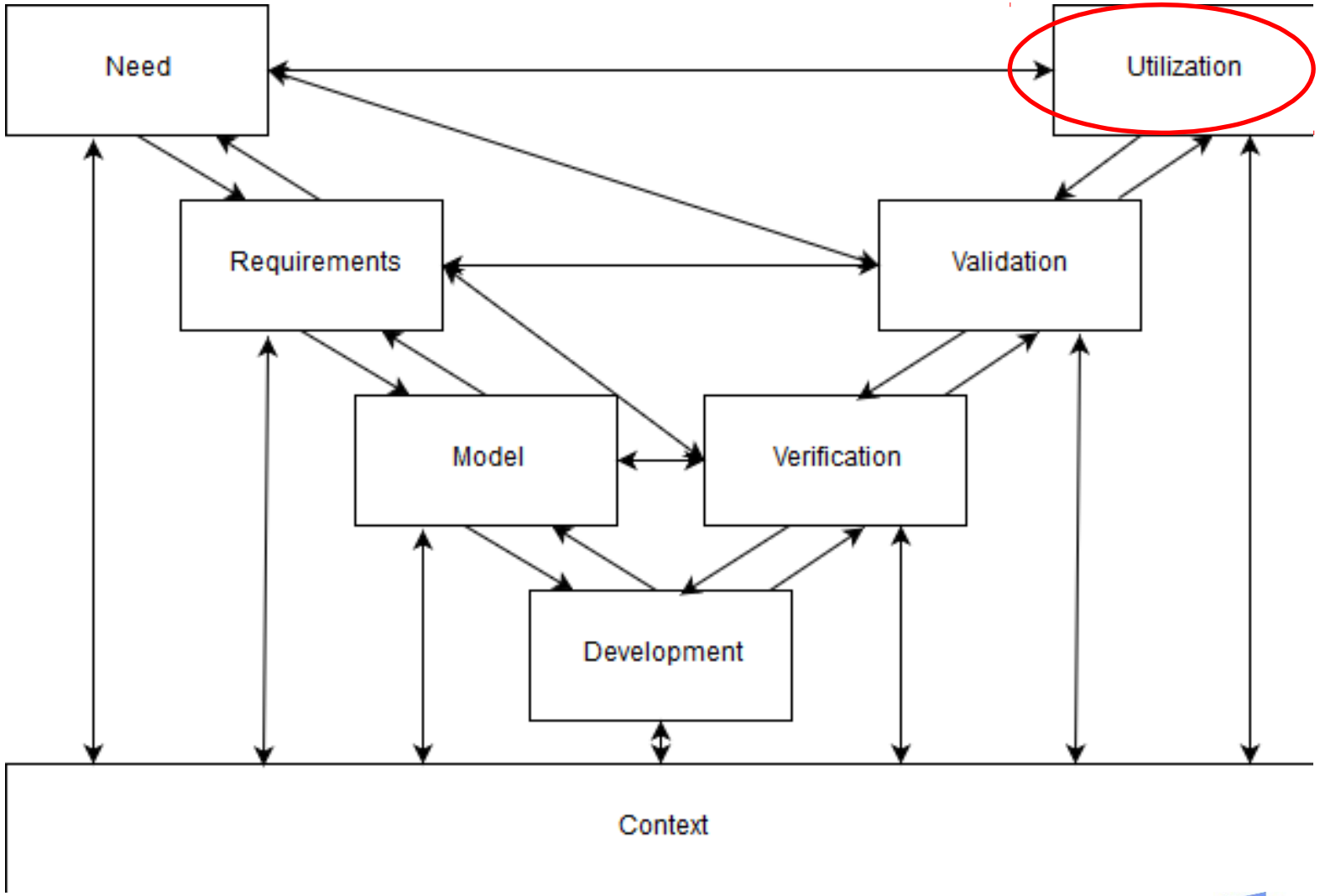
K Model


Need
Utilization
Requirements
Validation
Model
Verification
Development
Context

Validation

- Check if “the right entity has been built”
- The need must be met
- Validation of the requirements related to Real system environment
- Customer point of view
- Final product

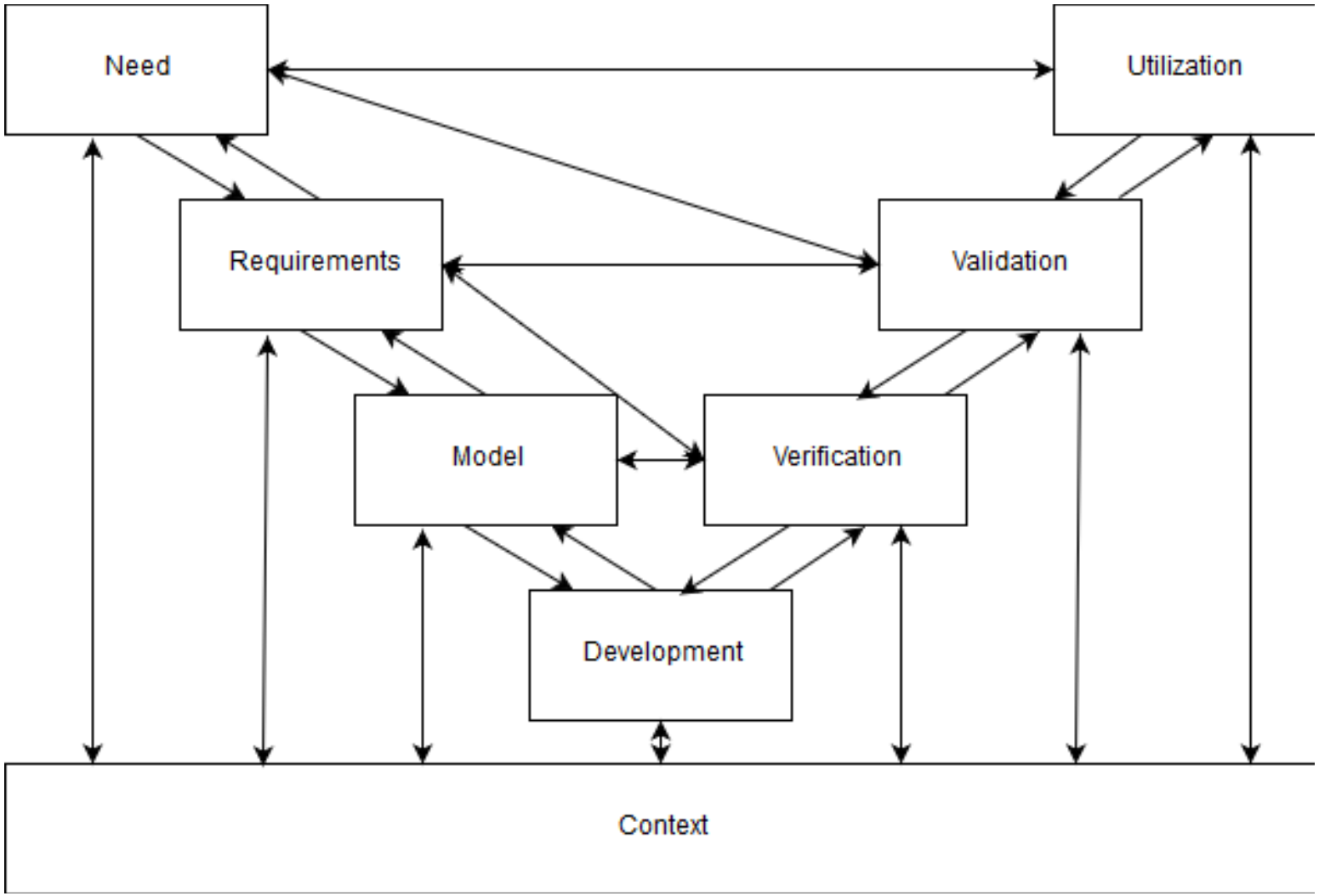
K Model


Need
Utilization
Requirements
Validation
Model
Verification
Development
Context

Utilization

- Trust@run.time
- Dynamicity of IoT must face with devices that (Join, Stay, Leave) the System

K Model (links)



Architecture Framework

- Introduction
  - Trust
  - Internet of Things
- Architecture Framework
  - K Model
  - Transversal Activities
- Use Case Scenario
- Conclusion

Transversal Activities

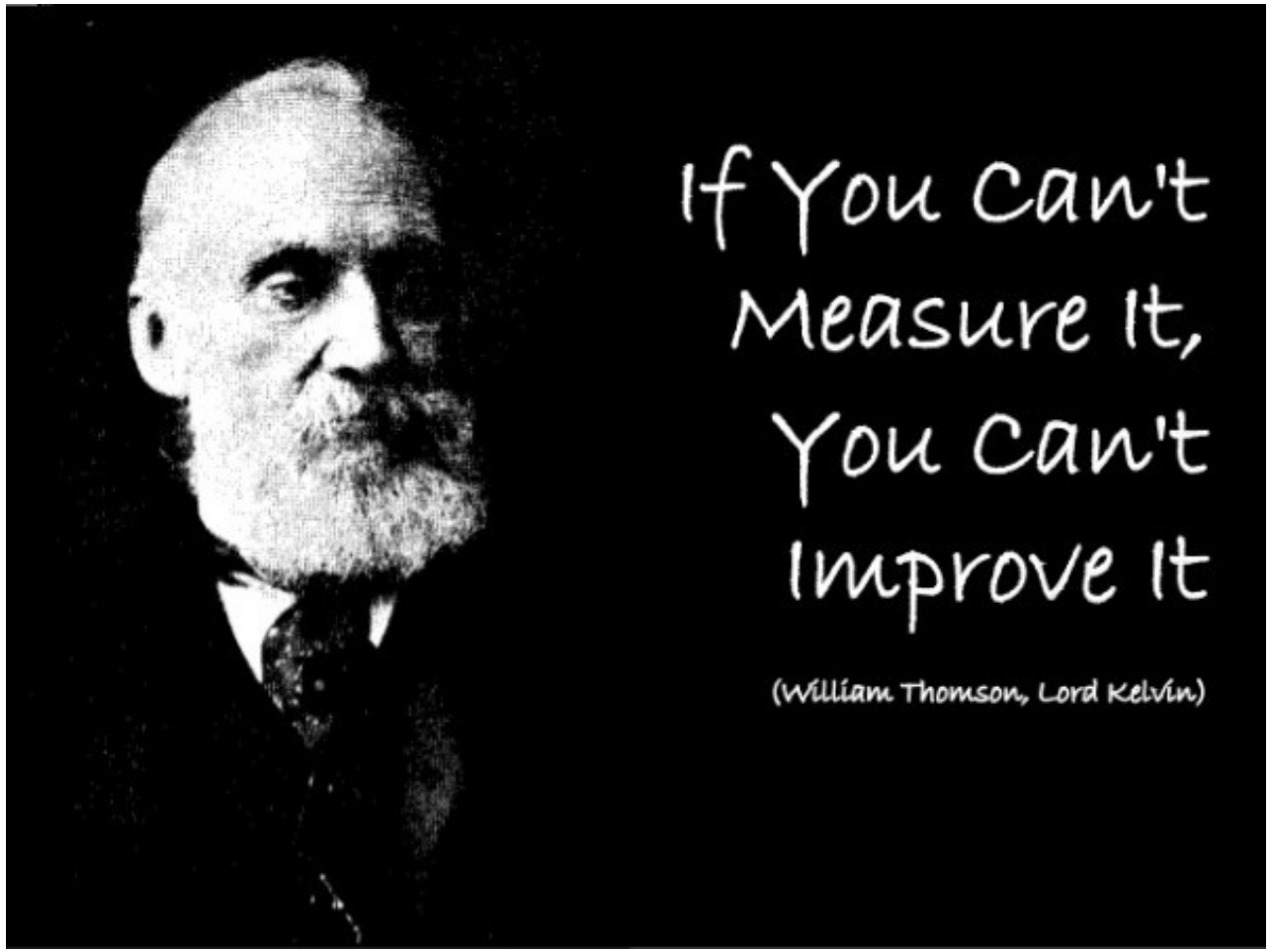
- Documentation
- Metrics
- Decision Gates
- Traceability
- Threat Analysis
- Risk Management
- Decision-Making

Documentation

- Connection
- Justification
- Procedures
- Guide
- *“Verba volant, scripta manent”*

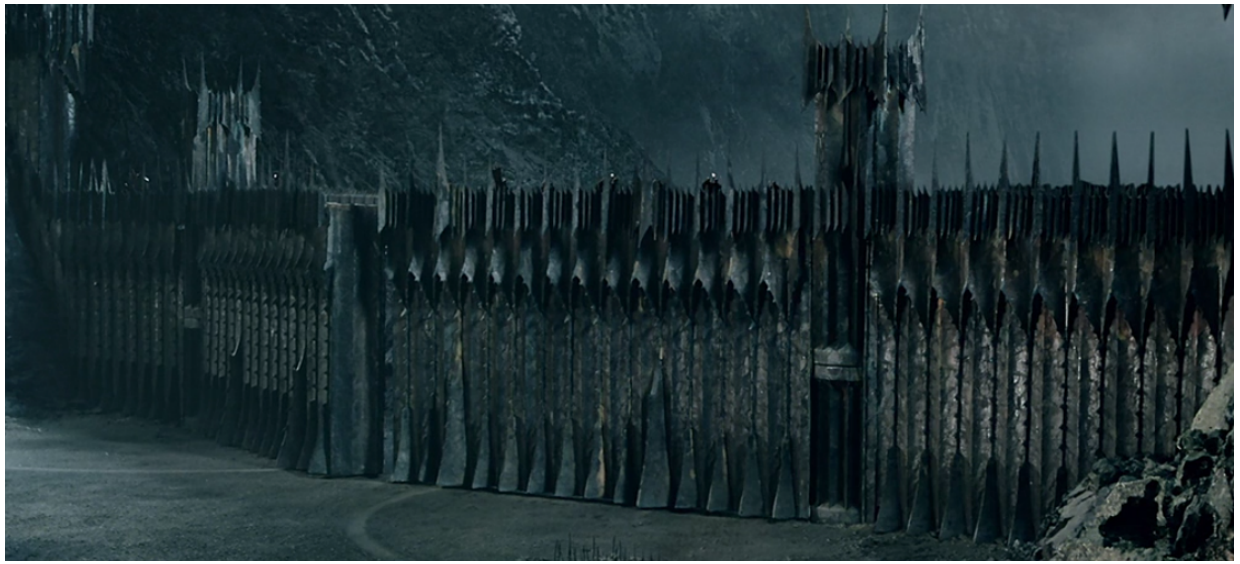
Metrics

- Trust Metrics
- Performance
- Efficiency
- Measures



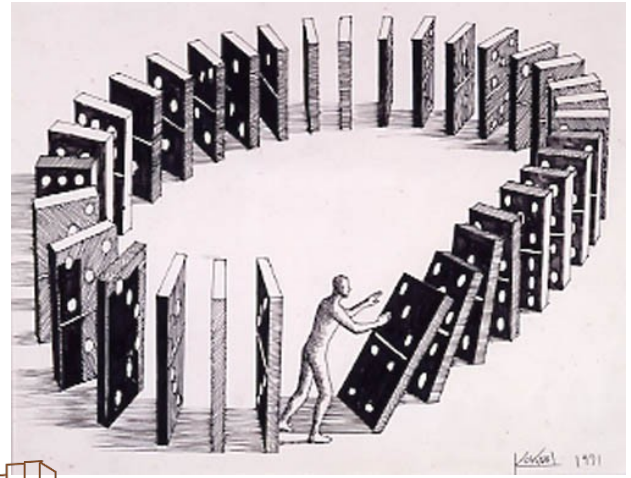
Decision Gates

- They permit to move between phases
- Back-Up in case something goes wrong



Traceability

- Connection between
  - Phases
  - Requirements
  - Activities and Phases
- Control Domino effects
- Help against Unintended Consequences

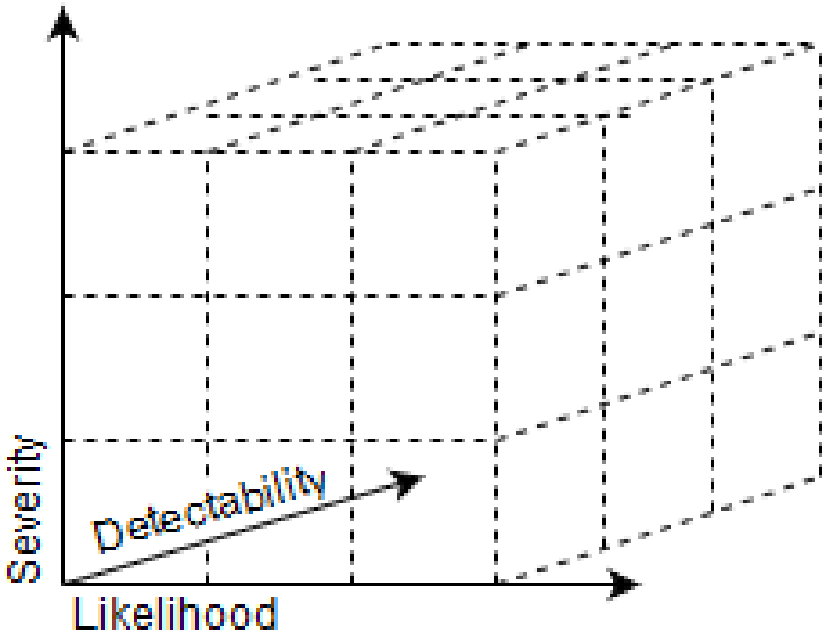


Threat Analysis

- Attacks
  - Internal
  - External
- Malfunctions
- Malwares

Risk Management

- Likelihood
- Severity
- Detectability



Decision Making

- Connected to many phases
  - Requirement
  - Model
  - Development
  - Utilization

Use Case Scenario

- Introduction
  - Trust
  - Internet of Things
- Architecture Framework
  - K Model
  - Transversal Activities
- Use Case Scenario
- Conclusion

Smart Cake Machine

Smart Hub

Smart Supermarket

Smart Fridge

Smart Cake Machine

Smart Cake Machine

- Context
  - Smart Home
  - Trusted Smart entities
- Need
  - Smart Cake Machine
- Requirements
  - Security Requirement
  - Trust Requirement
  - Usability Requirement
- Models
  - Trusted ClassDiagram
  - Trusted RequirementDiagram

Smart Cake Machine

- Development
  - Top Down
- Verification
  - Verify the correct functionalities of the Smart Cake Machine
- Validation
  - Validate it in the cooperation with Smart Fridge and Smart Supermarkets
- Utilization
  - Join the Smart Home
  - Deal with join and leaving Smart devices

Conclusion

- Introduction
  - Trust
  - Internet of Things
- Architecture Framework
  - K Model
  - Transversal Activities
- Use Case Scenario
- Conclusion

Conclusion

- IoT has brought new security challenges
- Trust as a key
- Software, Security and System Engineering approach to ensure trust in an entity
- Trust and other security properties are included in the whole life cycle
- K-Model
- Transversal Activities

Future Work

- Validation of the Framework
- We will expand the phases of the framework
- Application to a real complex IoT scenario
- Application in an IoT System

Questions?

Thanks to the European Commission, NECS Project and to the university of Malaga for the opportunity given to me.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 675320.

This work reflects only the author's view and the Research Executive Agency is not responsible for any use that may be made of the information it contains.