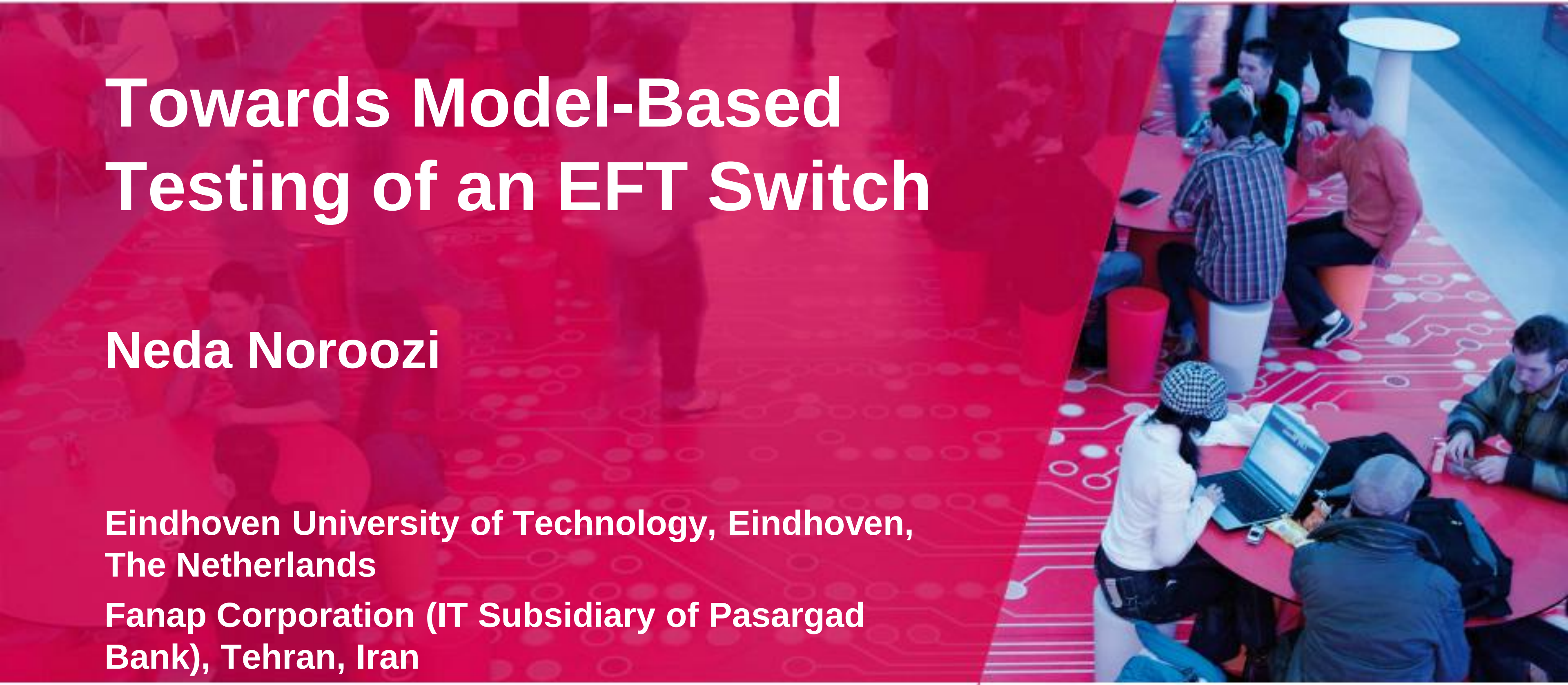


Towards Model-Based Testing of an EFT Switch

Neda Noroozi

Eindhoven University of Technology, Eindhoven, The Netherlands

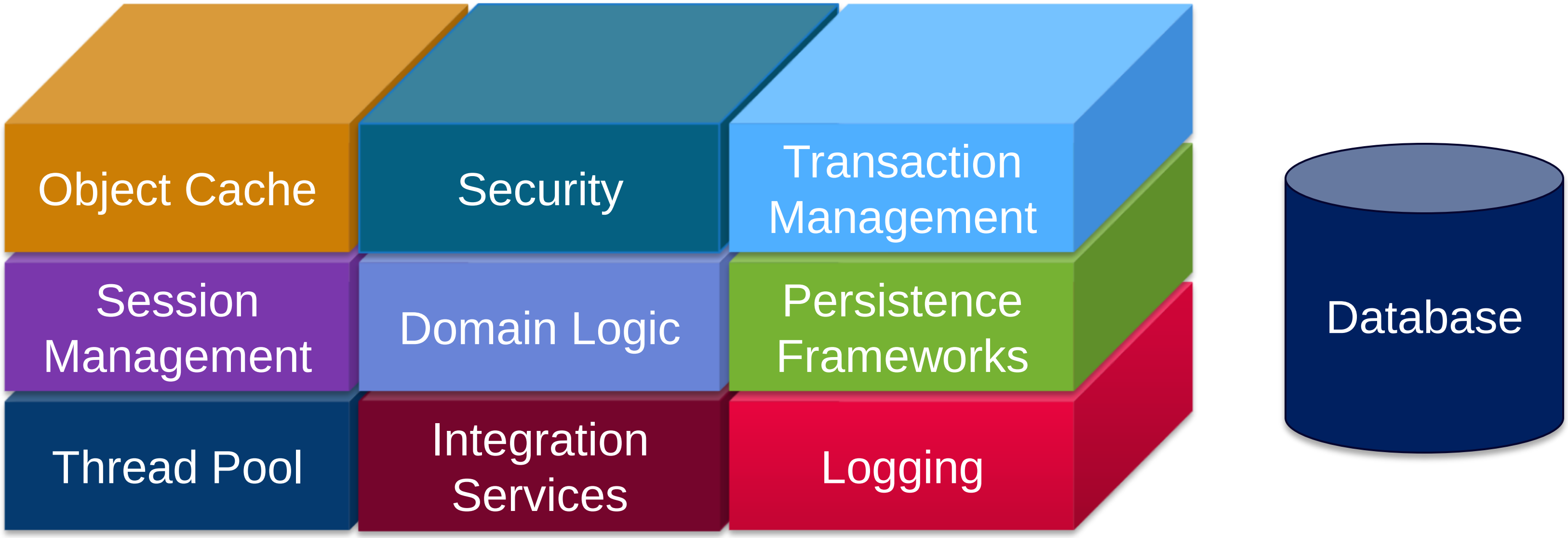
Fanap Corporation (IT Subsidiary of Pasargad Bank), Tehran, Iran

Technische Universiteit
Eindhoven
University of Technology

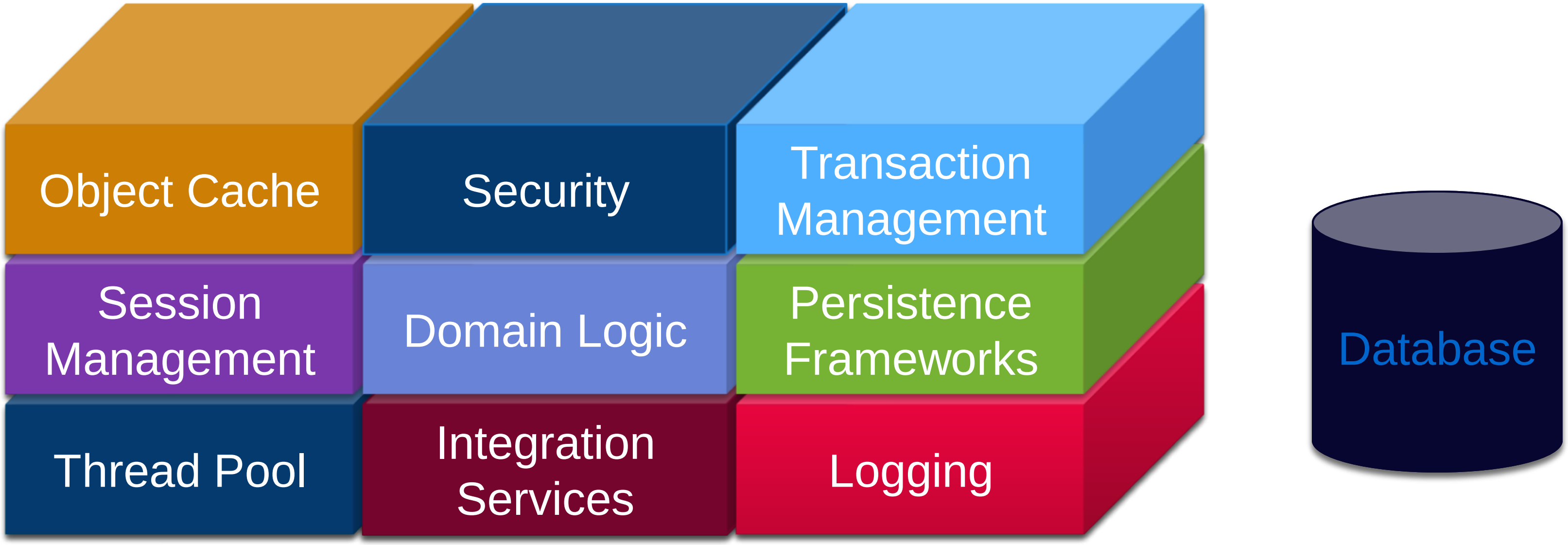
Where innovation starts

> *“ If you've got some trades going through at 10 milliseconds and some at 1 millisecond, that's a problem. Our customers don't like variance.”*
>
> Steve Rubinow (NYSE Group CTO)

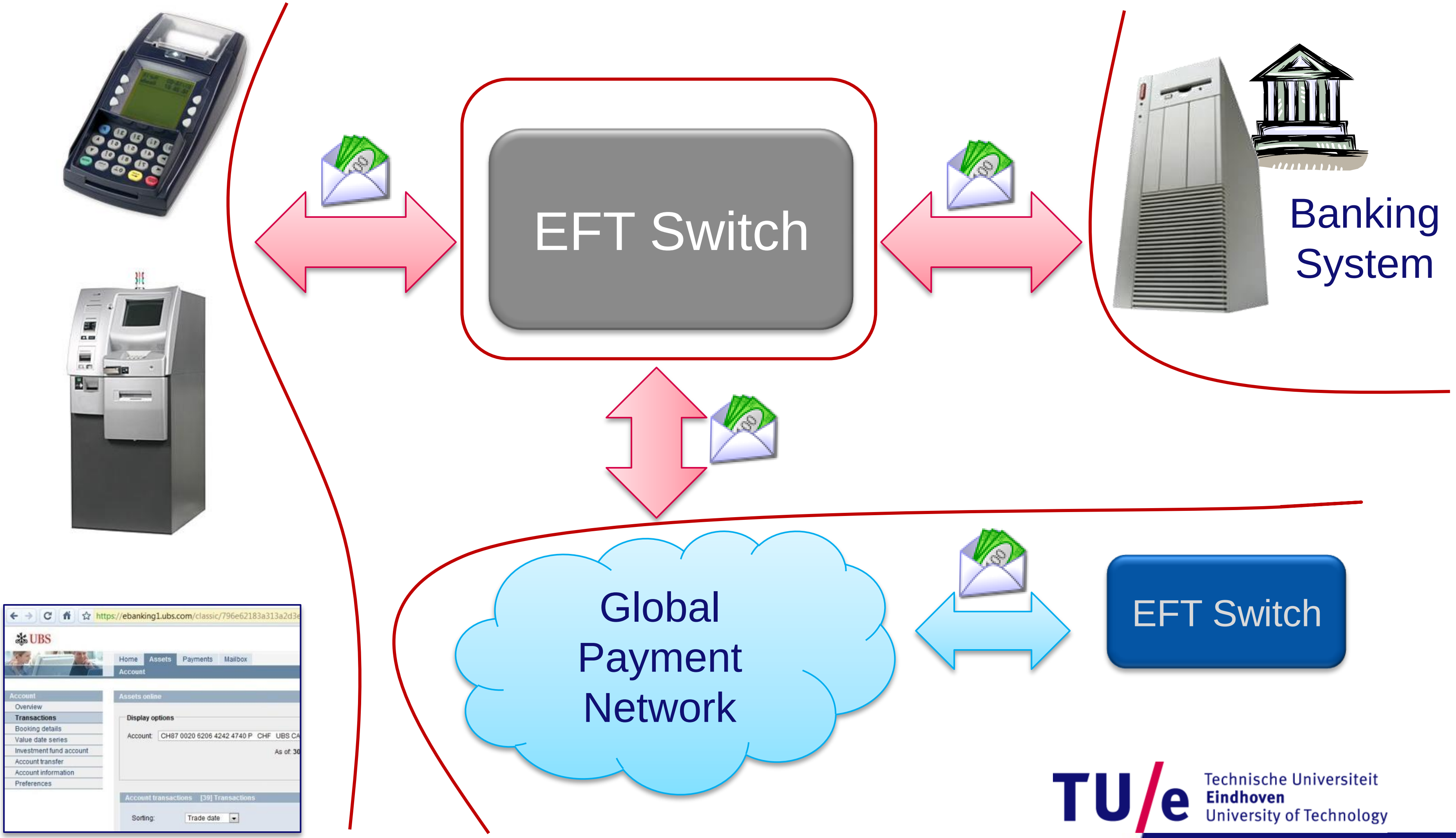
Overview- why testing?



Overview- why testing?



Overview- Electronic Funds Transfer



https://ebanking1.ubs.com/classic/796e62183a313a2d3e

Account: CH87 0020 6206 4242 4740 P CHF UBS CA

As of: 30

Sorting: Trade date

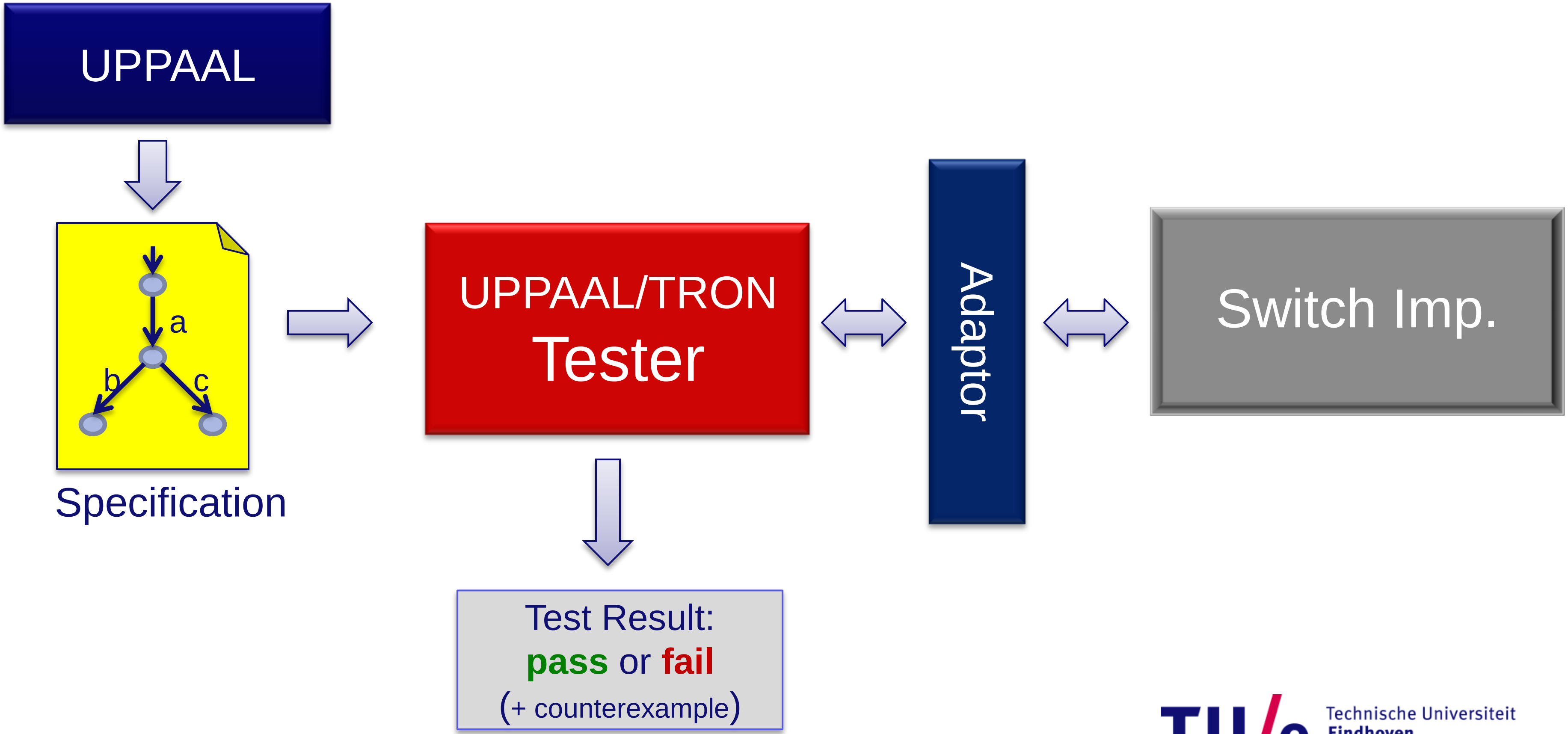
Overview- Desired qualities

- **Highly available**
- **Highly concurrent**
- **Highly secure**
- **High performance**
- **Highly extendable**

Specification

Informal	Specification	Formal
☺	Writing	☹
☹	Test case generation	☺

First attempt

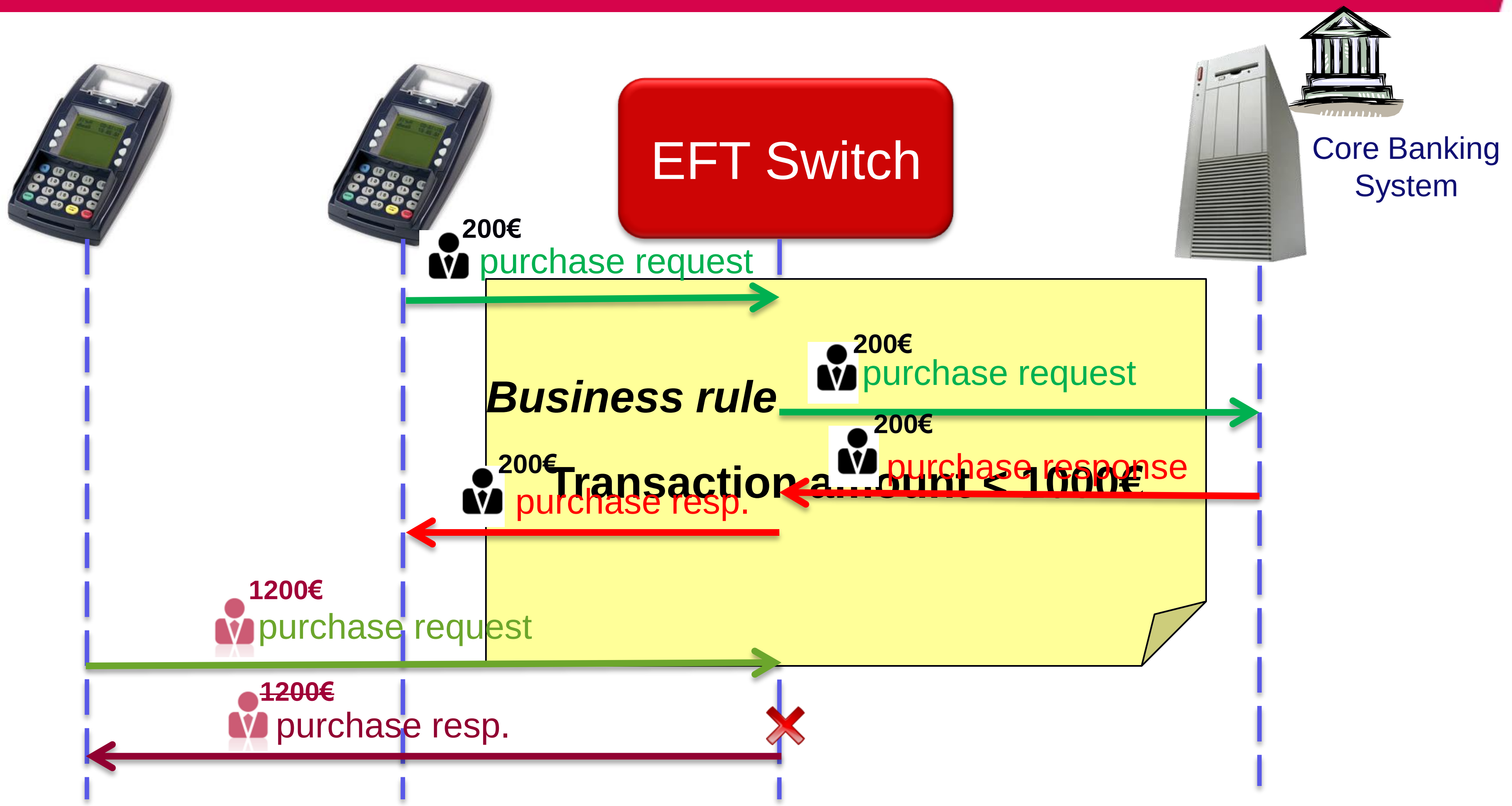


Test Result:
pass or **fail**
(+ counterexample)

First result

- **Modeling is time-consuming**
- **Not all possible scenarios is generating**

Data on transaction



EFT Switch

Core Banking System

200€ purchase request

Business rule

Transaction amount < 1000€

200€ purchase request

200€ purchase response

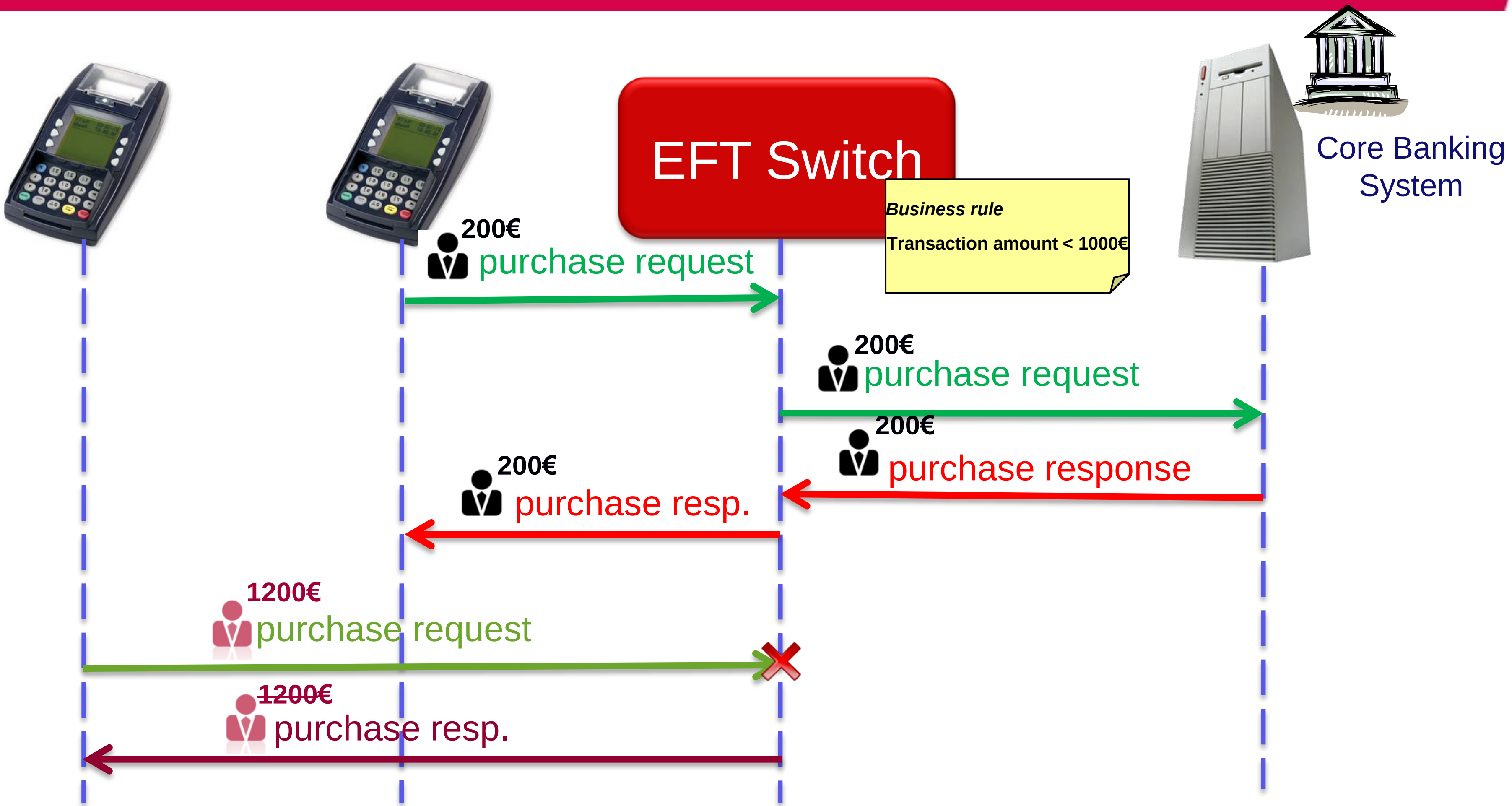
200€ purchase resp.

1200€ purchase request

~~1200€~~ purchase resp.

Technische Universiteit Eindhoven University of Technology

Concurrent transaction



EFT Switch

Core Banking System

Business rule

Transaction amount < 1000€

200€ purchase request

200€ purchase request

200€ purchase response

200€ purchase resp.

1200€ purchase request

~~1200€~~ purchase resp.

Asynchronous transaction

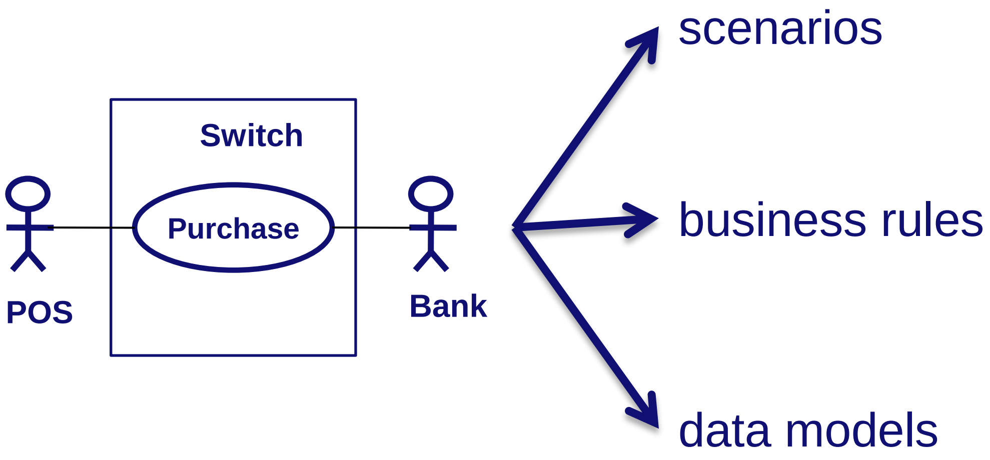
The theory of testing is mostly devoted to *synchronous* testing.

Motivations

- ❑Creating formal specification
- ❑Modeling data model
- ❑Running test efficiently
- ❑Modeling asynchronicity

Our Current Approach- Creating formal specification

- **Uses Software architecture products**

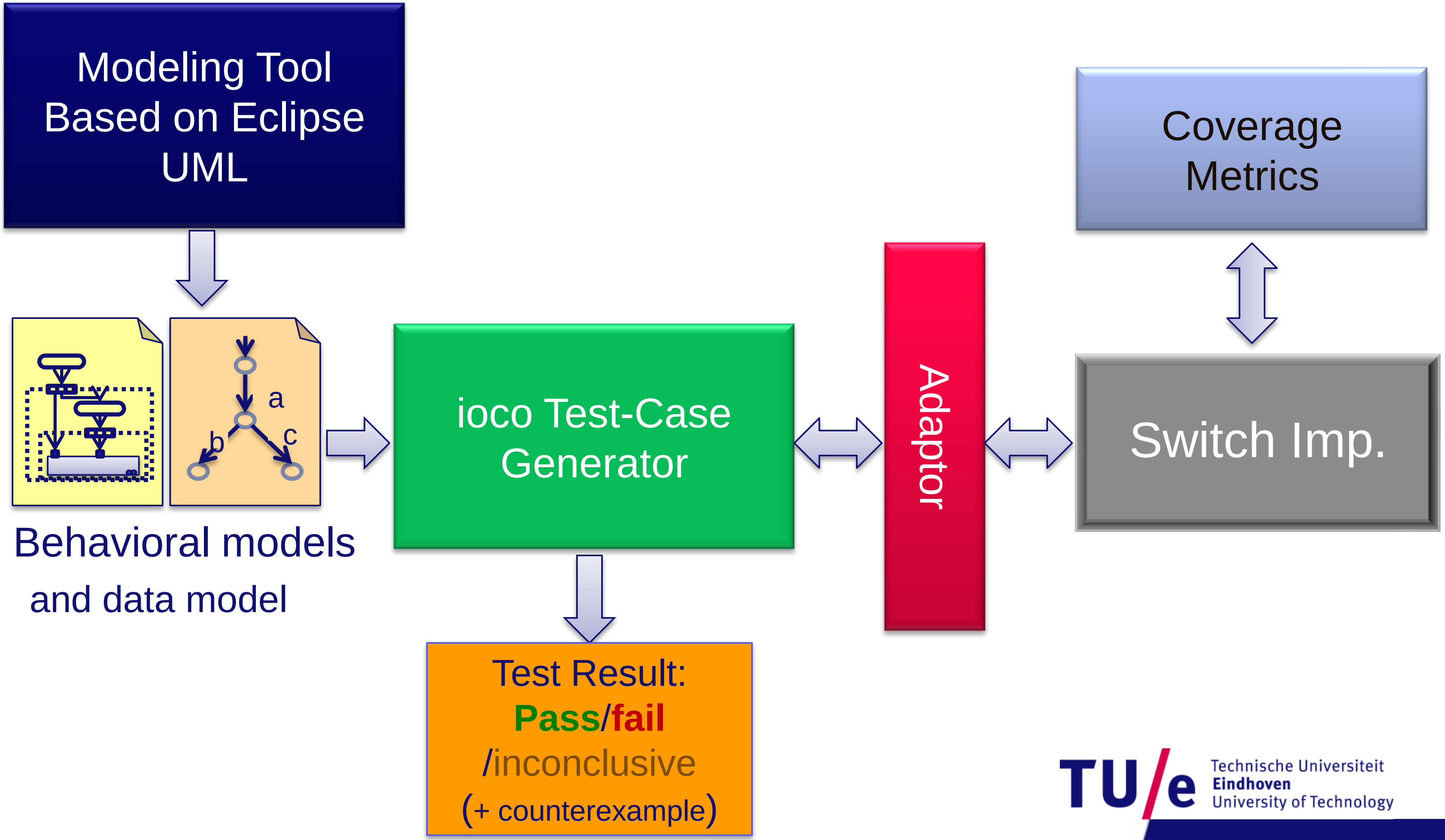


Our Current Approach- Modeling data and behavior

- Supports data modeling for test generation
  - Data types
  - Data relations
  - Data partitions

- Uses UML to express the models
  - System and environment behavior
  - Test Scenario

Implementation



Modeling Tool Based on Eclipse UML

Behavioral models and data model

ioco Test-Case Generator

Adaptor

Coverage Metrics

Switch Imp.

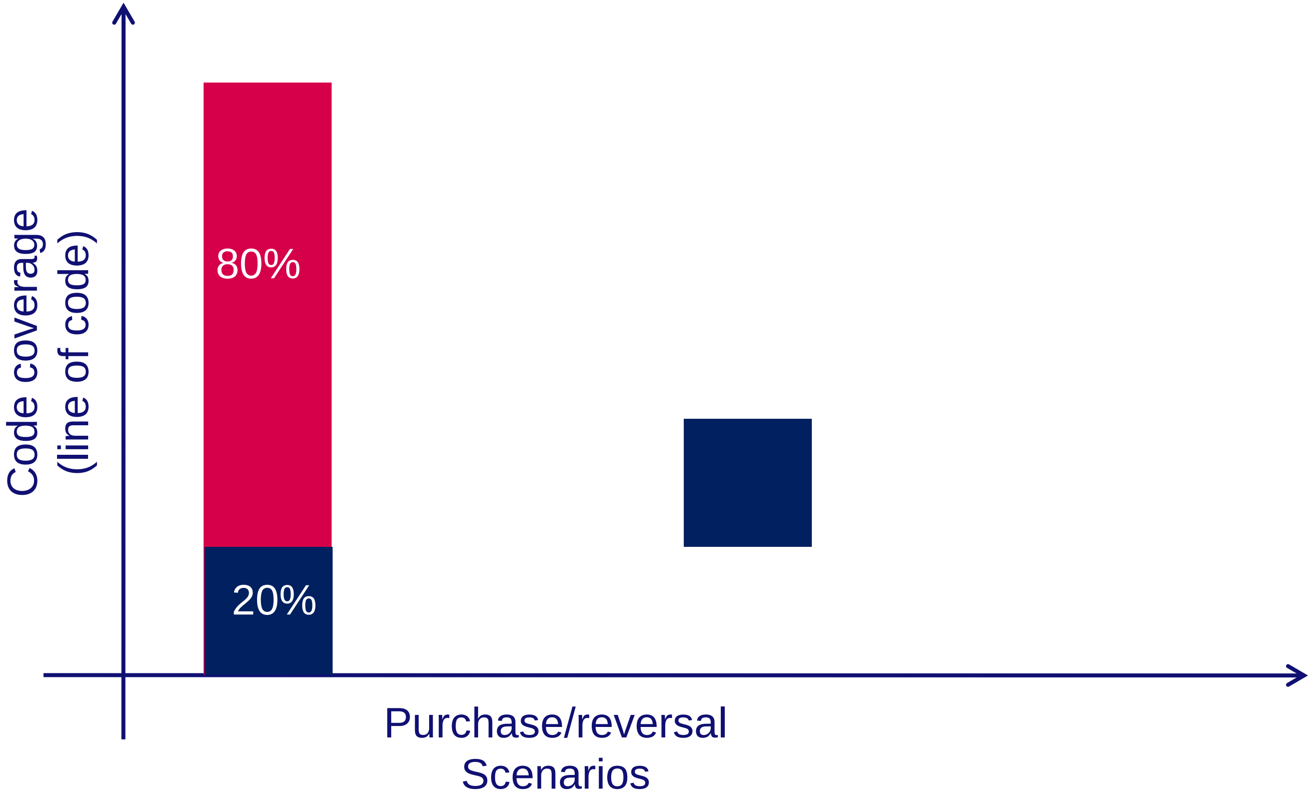
Test Result: **Pass**/**fail** /inconclusive (+ counterexample)

Achievements

- A method to generate formal specification
  - Behavioral model
  - Data model
- Managing test scenarios
  - Modeling the environment
  - Generating test case on-the-fly
  - Defining new goals
- Managing models

Achievements(cont.)

- **Data modeling**

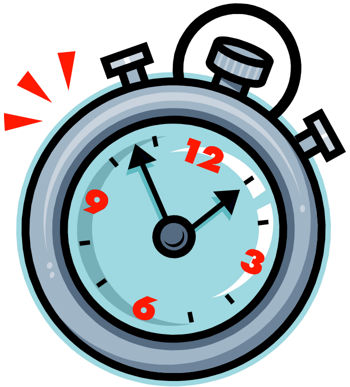


Results so far

- Have found a few bugs in the system
  - One traced back to null-pointer dereferencing
  - Poor exception handling
  - Problems in business rules
  - Resource pooling problems
  - A problem in concurrency

The Road Ahead

- **Checking database state**
- Avoiding a trace or a state
- Asynchronous testing
- Improving the modeling tool



THANK YOU

Technische Universiteit
Eindhoven
University of Technology