EXPLAINABILITY FIRST!

COUSTEAUING THE DEPTHS OF NEURAL NETWORKS

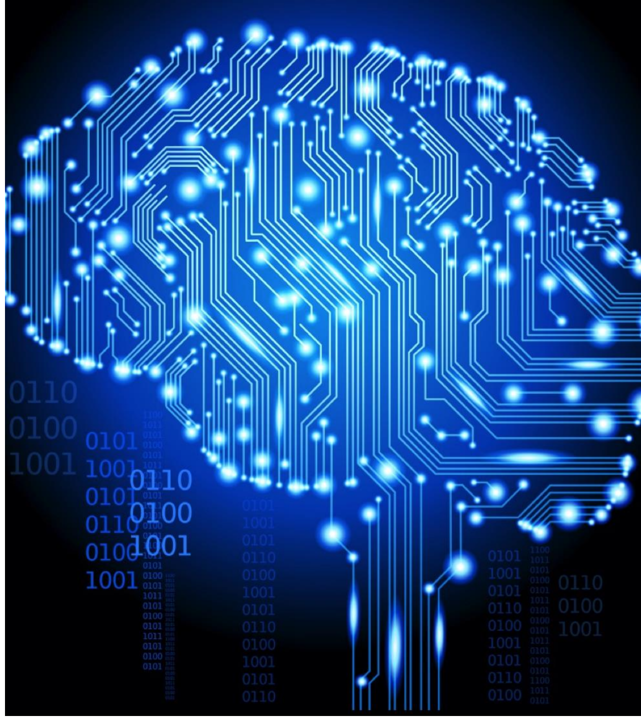
ES4CPS@Dagstuhl – Jan 7, 2019

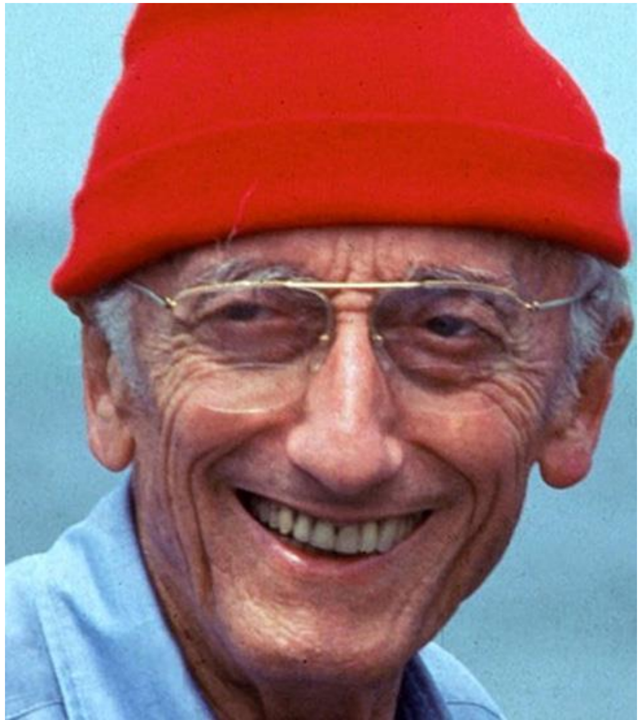
@mrksbrg

mrksbrg.com

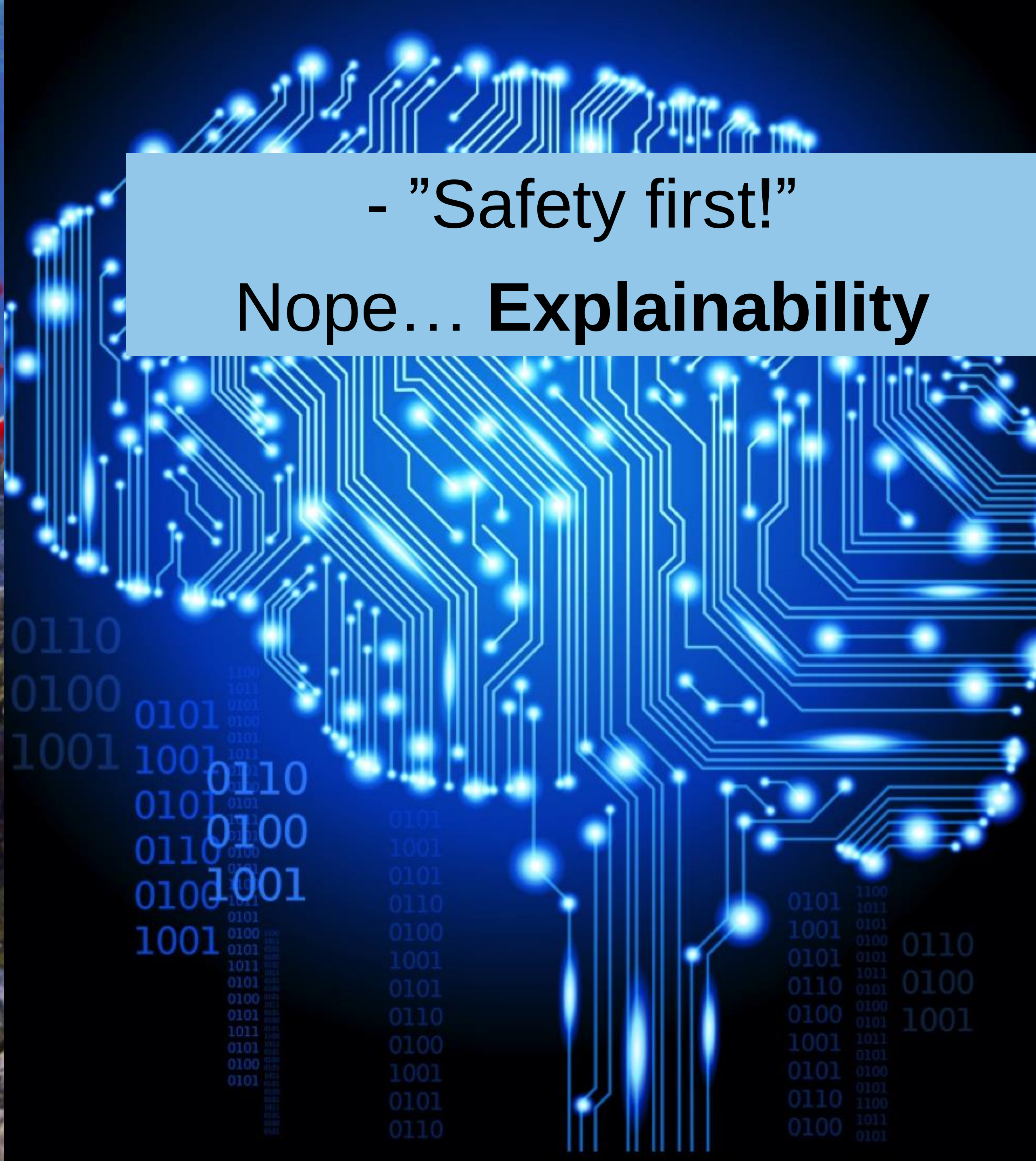
Markus Borg

RISE Research Institutes of Sweden AB





- “Aller voir!”



- ”Safety first!”

Nope… **Explainability**

Who is Markus Borg?

Development engineer, ABB, Malmö, Sweden 2007-2010

- Editor and compiler development
- Safety-critical systems

PhD student, Lund University, Sweden 2010-2015

- Machine learning for software engineering
- Bug reports and traceability

Senior researcher, RISE AB, Lund, Sweden 2015-

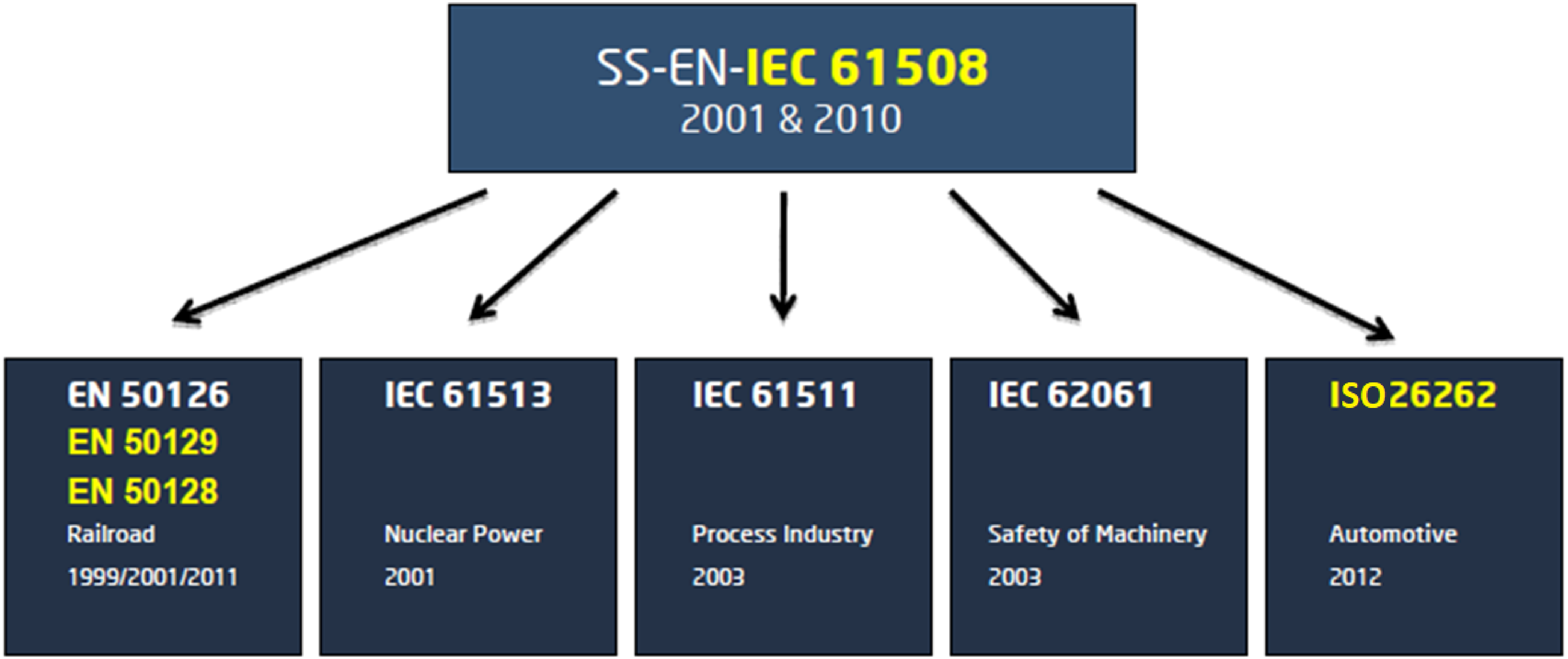
- Software engineering for machine learning
- Software testing and V&V

RISE

LUND UNIVERSITY

Background

Functional Safety Standards



SS-EN-IEC 61508
2001 & 2010

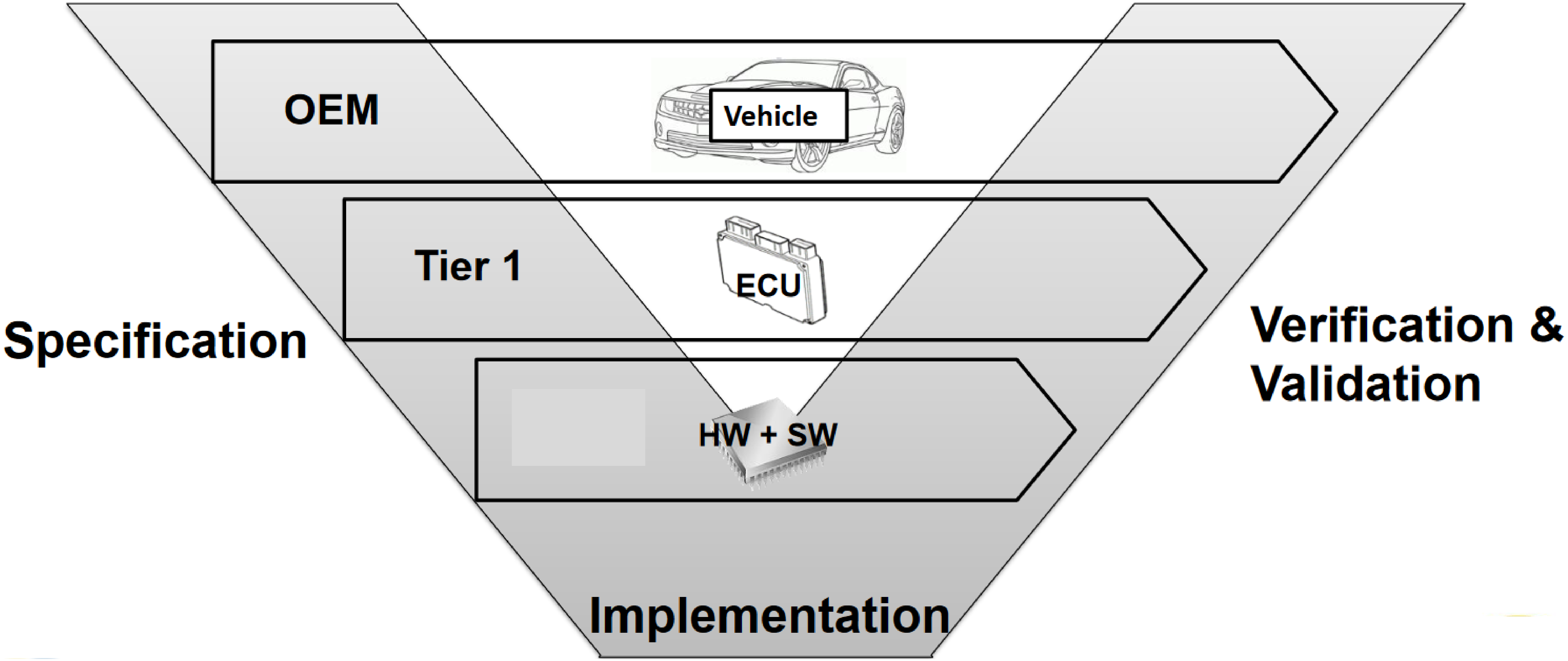
EN 50126
EN 50129
EN 50128
Railroad
1999/2001/2011

IEC 61513
Nuclear Power
2001

IEC 61511
Process Industry
2003

IEC 62061
Safety of Machinery
2003

ISO26262
Automotive
2012



Specification

OEM

Vehicle

Tier 1

ECU

HW + SW

Implementation

Verification & Validation

Achieving Safety in Software Systems

1. Develop understanding of situations that lead to safety-related failures
   - Hazard = system state that could lead to an accident
2. Design software so that such failures do not occur
   - Fault tree analysis

The system shall never hurt anyone
- even if the system does not conform to its specification

Safety certification => Put evidence on the table!

- Safety requirement: “Stop for crossing pedestrians”
- How do you argue in the safety case?

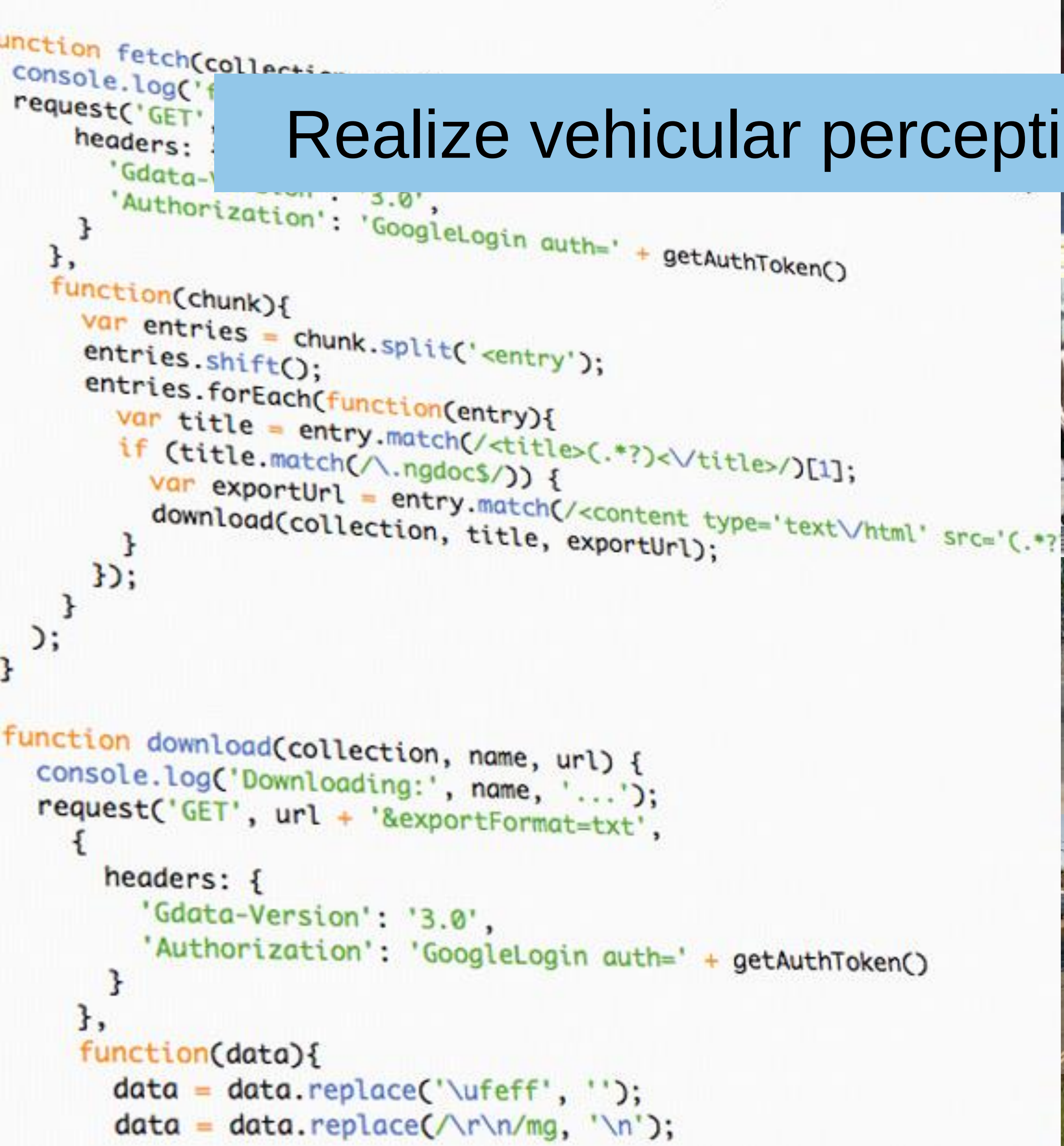
Safety evidence – In a nutshell

- System specifications
  - and why we believe it is valid
- Comprehensive V&V process descriptions
  - and its results
  - coverage testing for all critical code
- Software process descriptions
  - hazard register and safety requirements
  - code reviews
  - traceability from safety requirements to code and tests
  - …

Application context

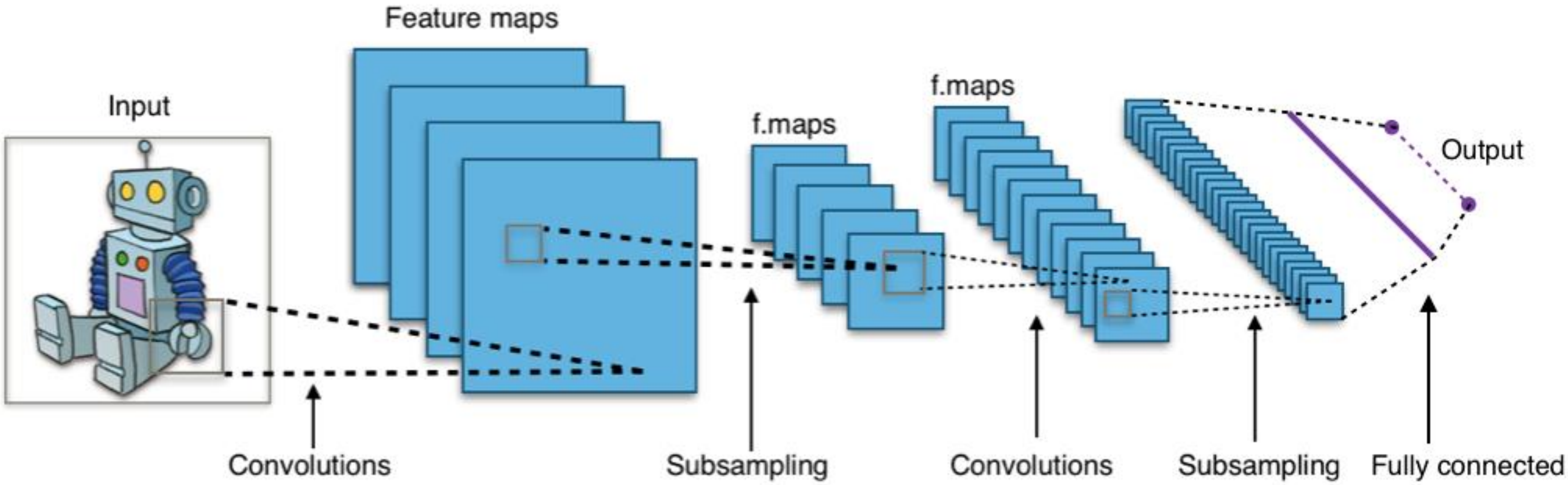
Safe-Req-A1:
In autonomous highway mode A, the vehicle shall keep a minimum safe distance of 50 m to preceding traffic

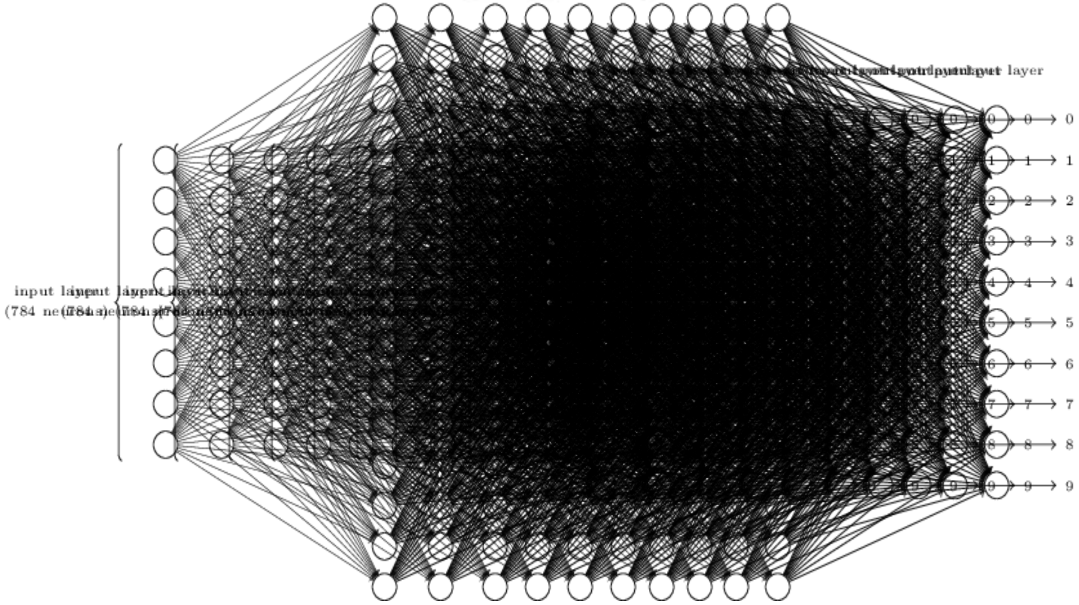
Realize vehicular perception using deep learning



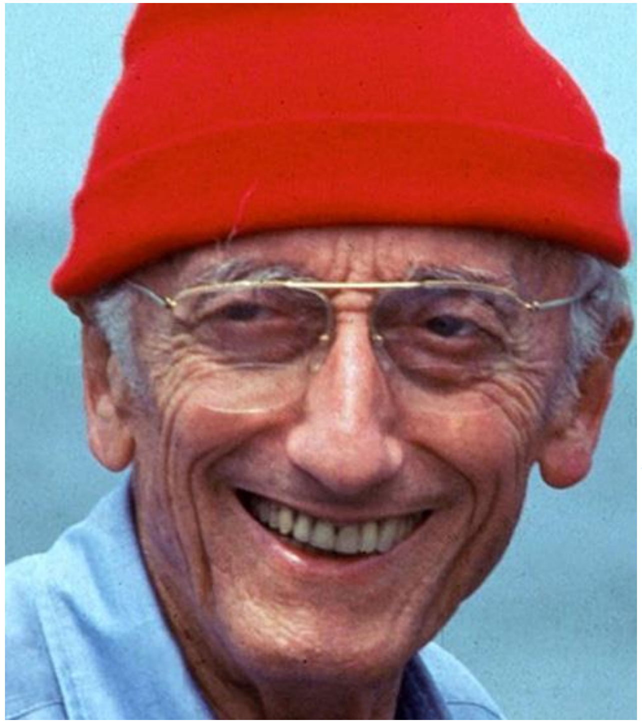
```
unction fetch(collecti
  console.log('f
  request('GET'
      headers: 
        'Gdata-V        : '3.0',
        'Authorization': 'GoogleLogin auth=' + getAuthToken()
      }
    },
    function(chunk){
      var entries = chunk.split('<entry');
      entries.shift();
      entries.forEach(function(entry){
        var title = entry.match(/<title>(.*?)<\/title>/)[1];
        if (title.match(/\.ngdoc$/)) {
          var exportUrl = entry.match(/<content type='text\/html' src='(.*?
          download(collection, title, exportUrl);
        }
      });
    }
  );
}

function download(collection, name, url) {
  console.log('Downloading:', name, '...');
  request('GET', url + '&exportFormat=txt',
    {
      headers: {
        'Gdata-Version': '3.0',
        'Authorization': 'GoogleLogin auth=' + getAuthToken()
      }
    },
    function(data){
      data = data.replace('\ufeff', '');
      data = data.replace(/\r\n/mg, '\n');
```

Autonomous Driving thanks to Convolutional Neural Networks





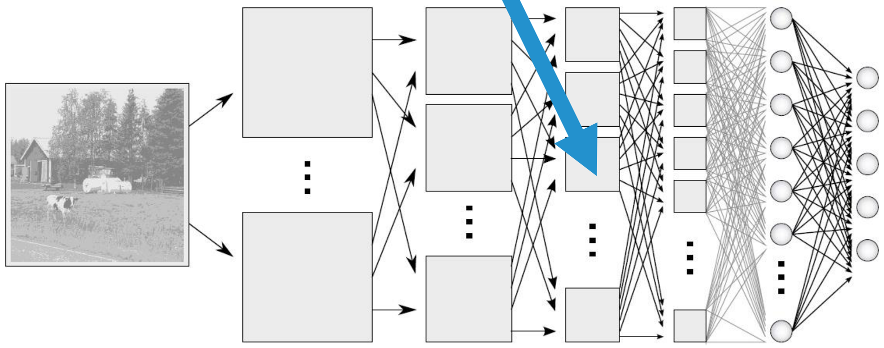
Trace from Safe-Req-A1 to... what?



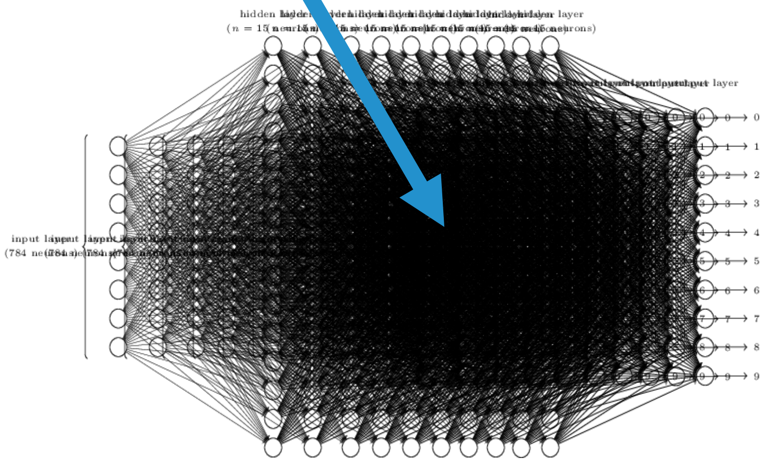
”Aller voir!”

Trace from Safe-Req-A1 to… what?

1) inside a human-interpretable model of a deep neural network



2) parameter values in a trained deep learning model



3) in training examples used to train and test the deep learning model

Open challenge

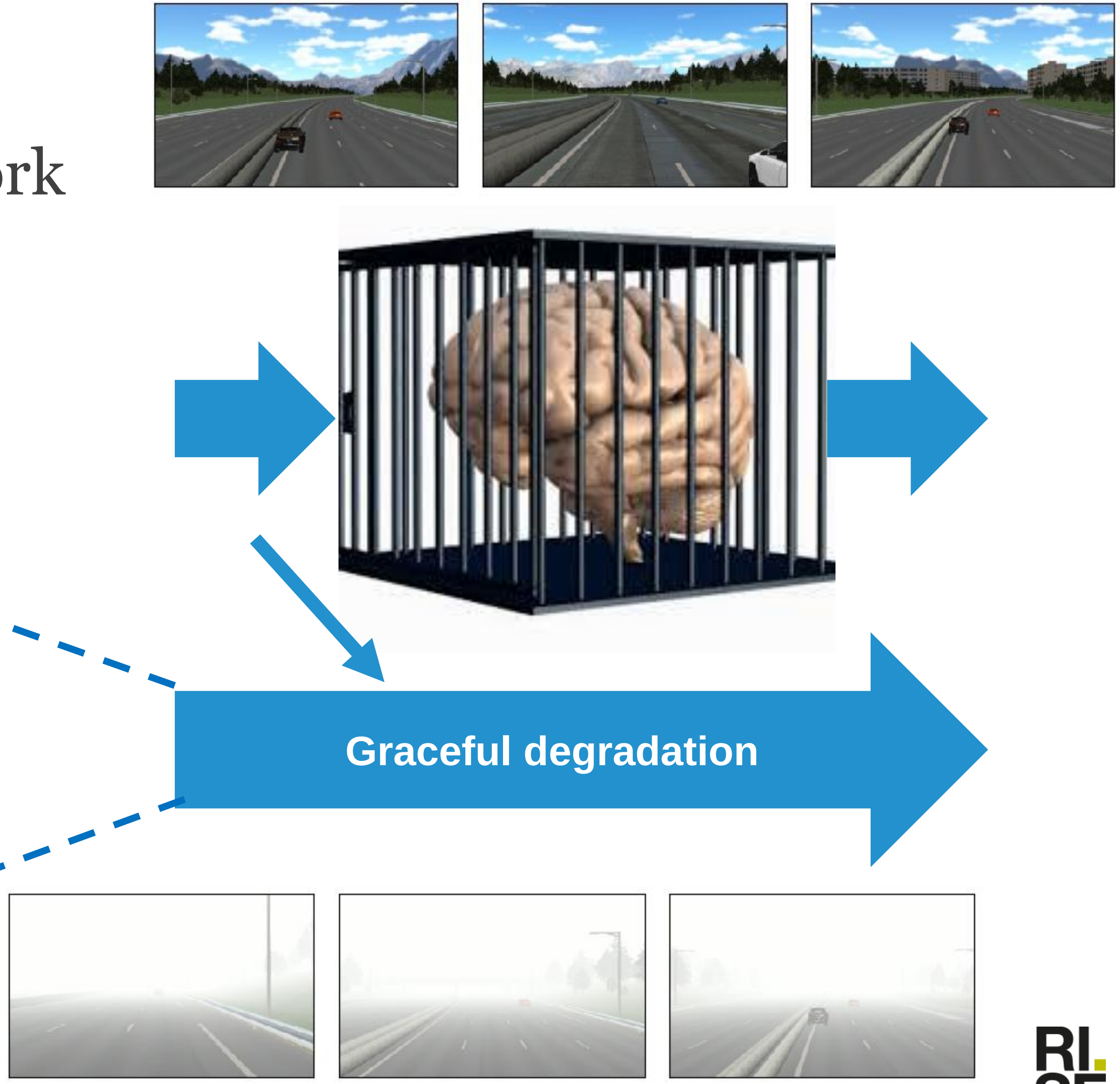
System feature - Autonomous highway driving

- FR1: … shall have an autonomous mode … in normal conditions…
- FR2: If the conditions change … shall request manual mode …
- FR3: If the driver does not comply … perform graceful degradation

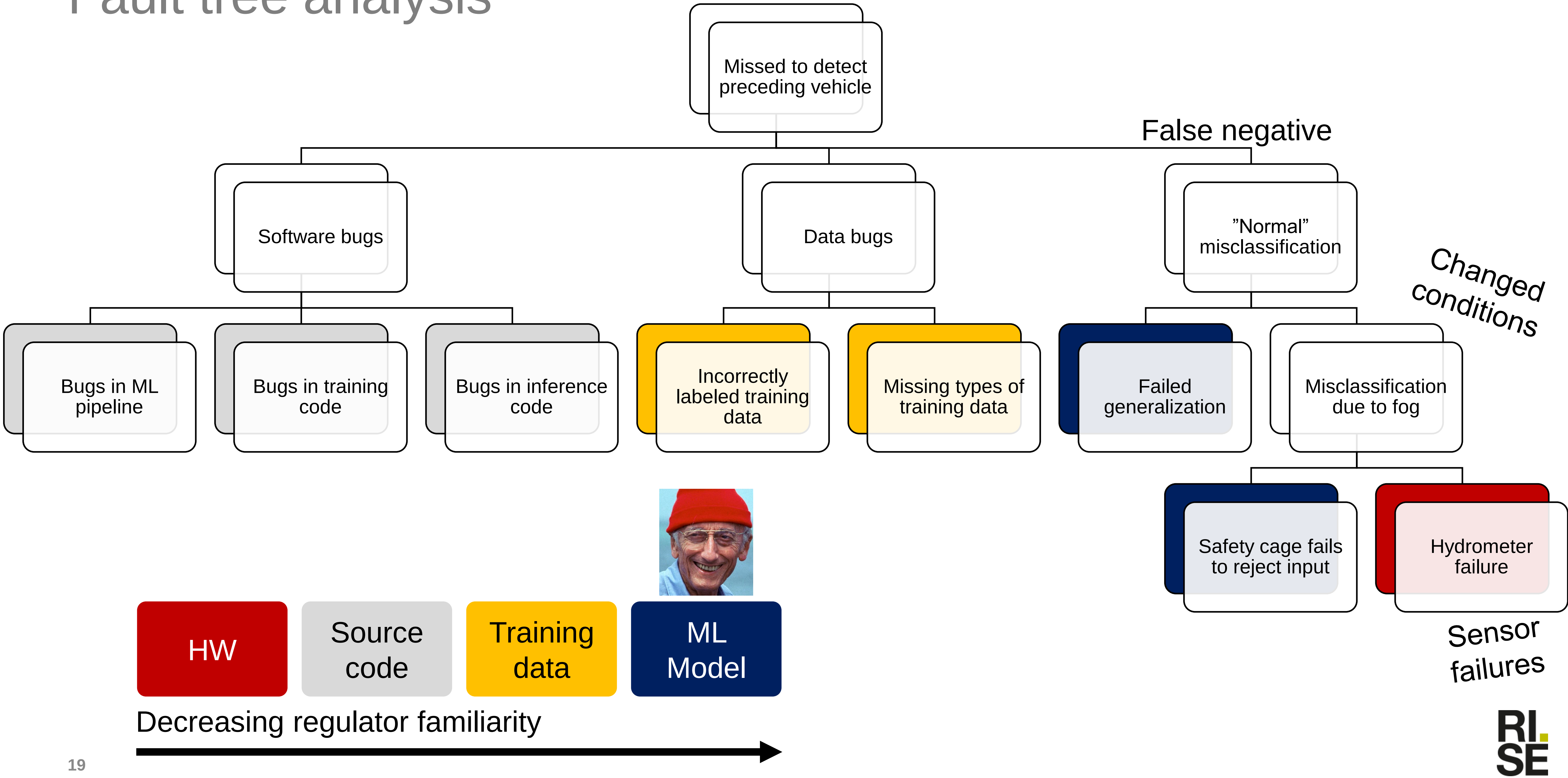
Safety cage architecture

- Add reject option for deep network
  - Novelty detection

- Graceful degradation
  - turn on hazard lights
  - slow down
  - attempt to pull over



Fault tree analysis



Missed to detect preceding vehicle

False negative

Software bugs

Data bugs

”Normal” misclassification

Changed conditions

Bugs in ML pipeline

Bugs in training code

Bugs in inference code

Incorrectly labeled training data

Missing types of training data

Failed generalization

Misclassification due to fog

Safety cage fails to reject input

Hydrometer failure

Sensor failures

HW

Source code

Training data

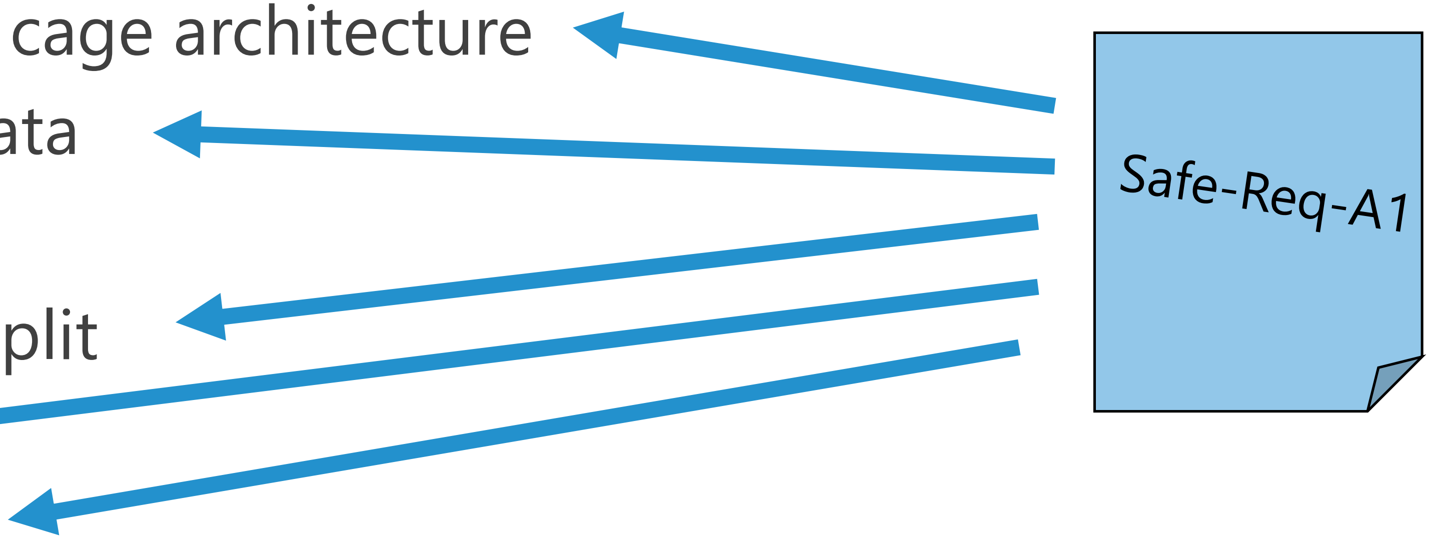
ML Model

Decreasing regulator familiarity

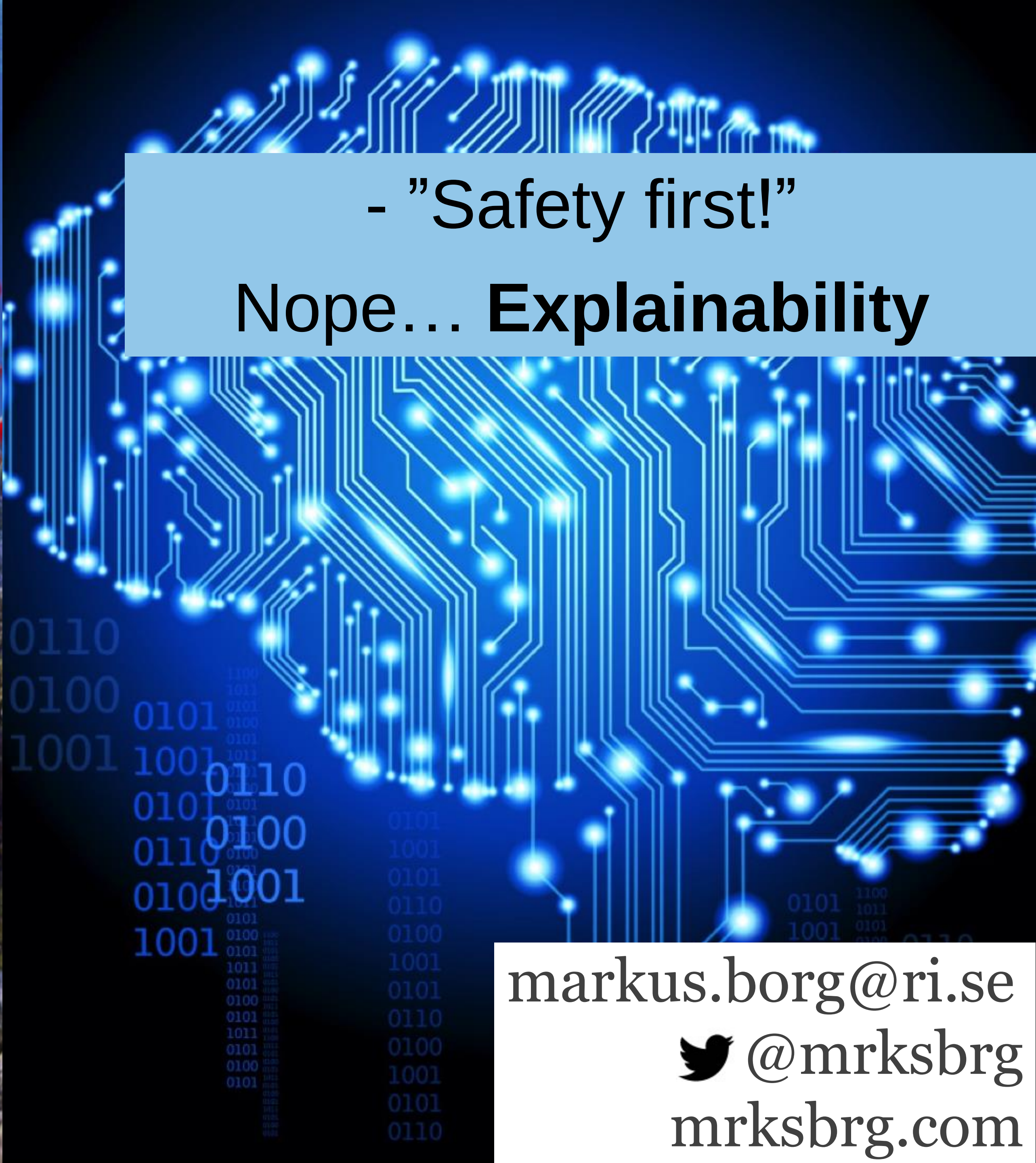
Explainability additions

POST-DEPL ARNING...

- System specifications
  - CNN architecture, safety cage architecture
  - description of training data
- V&V process descriptions
  - training-validation-test split
  - neuron coverage
  - approach to simulation
- Software process extensions
  - new ML hazards advarsarial example mitigation strategy
  - traceability from all safety requirements to **data** and **code** and **tests**
  - staff ML training



- “Aller voir”



- ”Safety first!”

Nope… **Explainability**

markus.borg@ri.se

@mrksbrg

mrksbrg.com