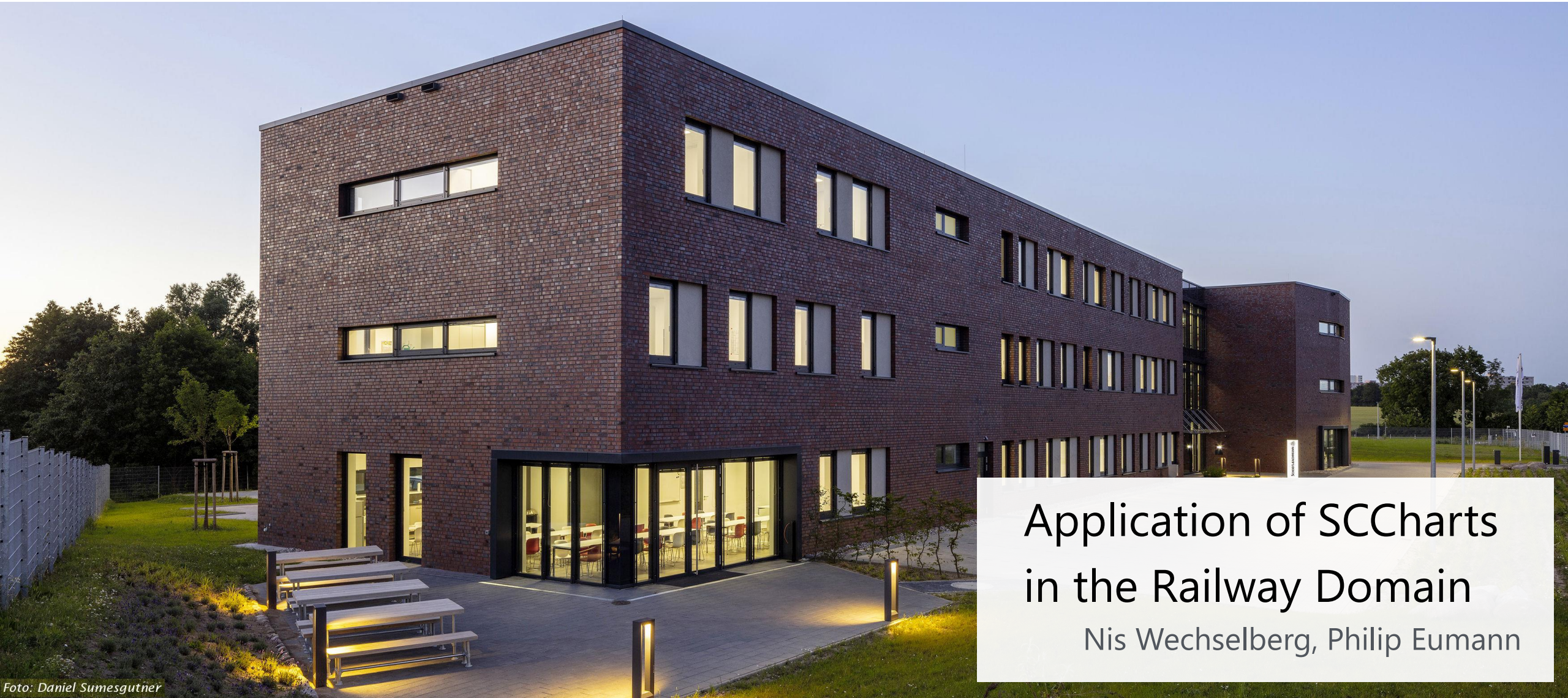
Welcome to SCHEIDT&BACHMANN

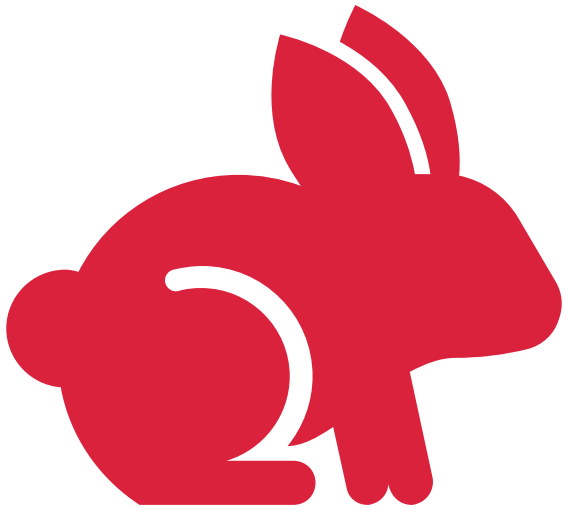


Application of SCCharts in the Railway Domain

Nis Wechselberg, Philip Eumann

Foto: Daniel Sumesgutner

/ Two Main Challenges in the Railway Sector



/ Safety?

by Nxr-at – own work, CC BY-SA 4.0,
https://commons.wikimedia.org/w/index.php?curid=74649486

by Alupus – own work, CC BY-SA 3.0,
https://commons.wikimedia.org/w/index.php?curid=24722748

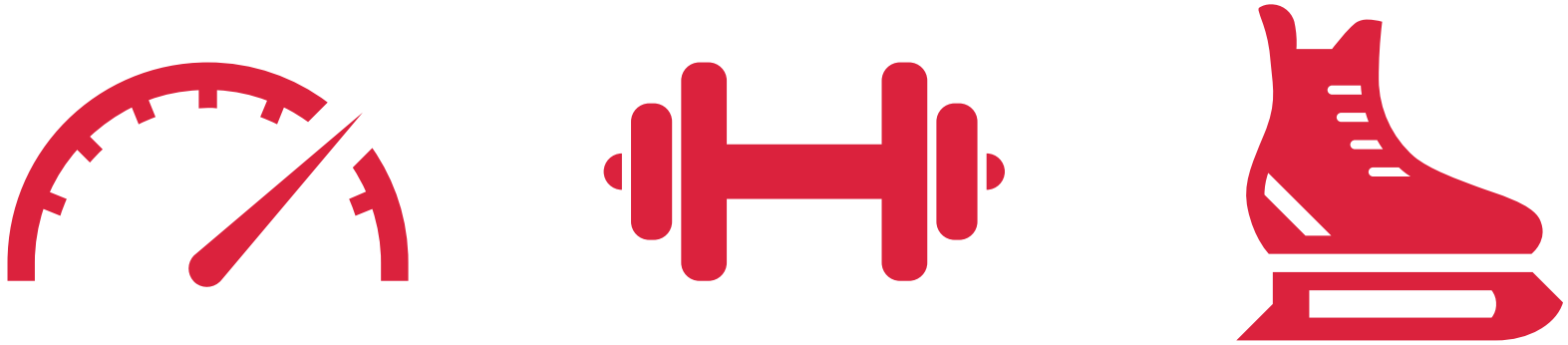
/ Safety?

by I, Sese Ingolstadt, CC BY-SA 2.5, https://commons.wikimedia.org/w/index.php?curid=1877619

/ Safety

/ Safety

/ Trains cannot be driven on sight



/ Two Core Mechanisms

- Signals can only turn green when the track beyond them is safe
  - Signal Dependency

by Olga Ernst & Hp.Baumeler – own work, CC BY-SA 4.0,
https://commons.wikimedia.org/w/index.php?curid=67973231

/ Two Core Mechanisms

- Trains can only pass green signals

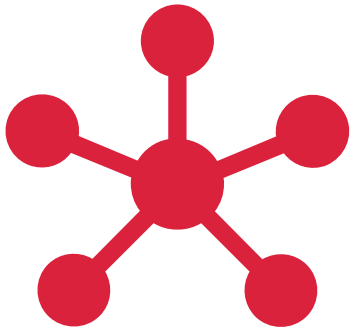
by Christian Liebscher – own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=39224581

CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=519699

/ Safety Requirements

- Supervision by Federal Railway Agency (Eisenbahnbundesamt, EBA)
- Mechanical components can easily be verified
- Local changes do not affect the entire system
- Changes and updates can be carried out by third-party technicians

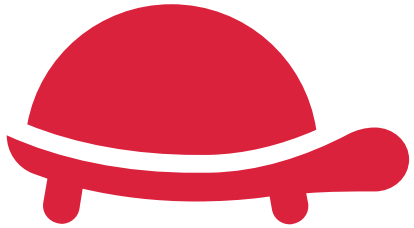
/ The Issue with Software

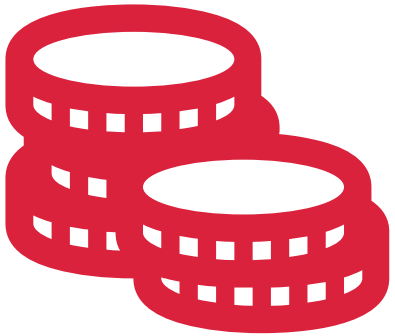


/ Standard Development Process

CENELEC
EUROPEAN COMMITTEE
FOR ELECTROTECHNICAL STANDARDIZATION

- Highest **S**afety and **I**ntegrity **L**evel 4 for electronic interlockings





*

/ Efficiency

by 2013-06-08_Highflyer_HP_L4729.JPG: Alchemist-hp (talk)derivative work: Hic et nunc – This file was derived from the following work: 2013-06-08 Highflyer HP L4729.JPG:, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=27038423

by Hbf878, OpenStreetMap contributors – own work, using OpenStreetMap data, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=89862775

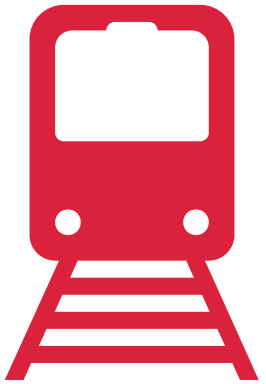
/ Efficiency

https://www.swr.de/swraktuell/baden-wuerttemberg/1690538908470%2Cbahnbaustelle-symbolbild-100~_v-16x7@2dL_-594eb175bf96444e7f86c89c3d9f78feed295e4a.jpg

/ Command and Control Systems

- Ensure efficient traffic flow
- High levels of automation
- Network-wide coordination

/ Command and Control Systems

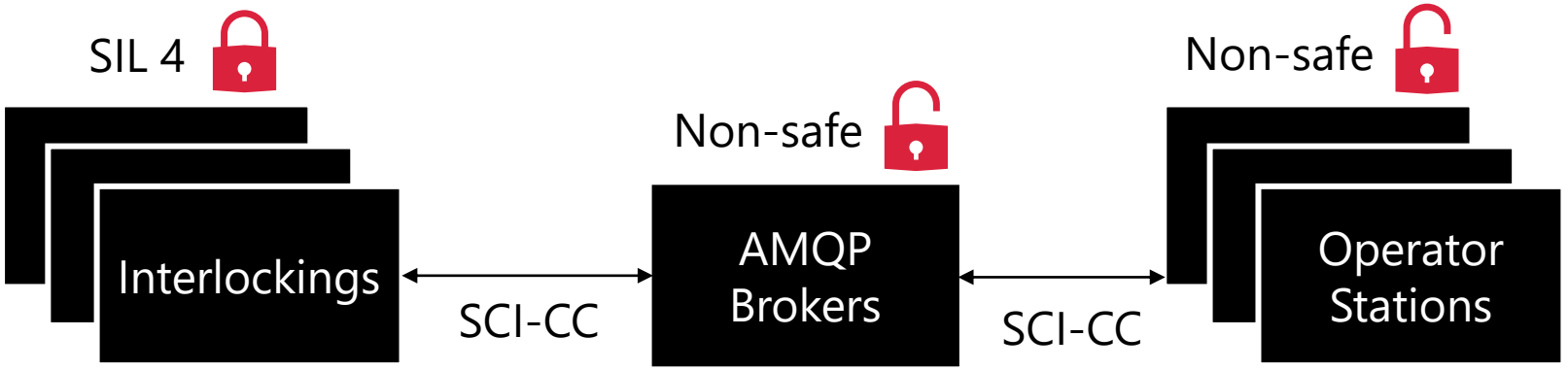


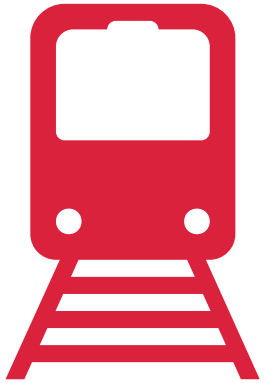
- Complex distributed systems
- Operators can override safety mechanisms

/ Command and Control Systems today

- Operator station is safety-critical system (SIL 2)
- to keep complexity low, other applications run on separate machines
- multiple sign-ins
- multiple input devices
- proprietary communication interfaces
- linked safety evaluation

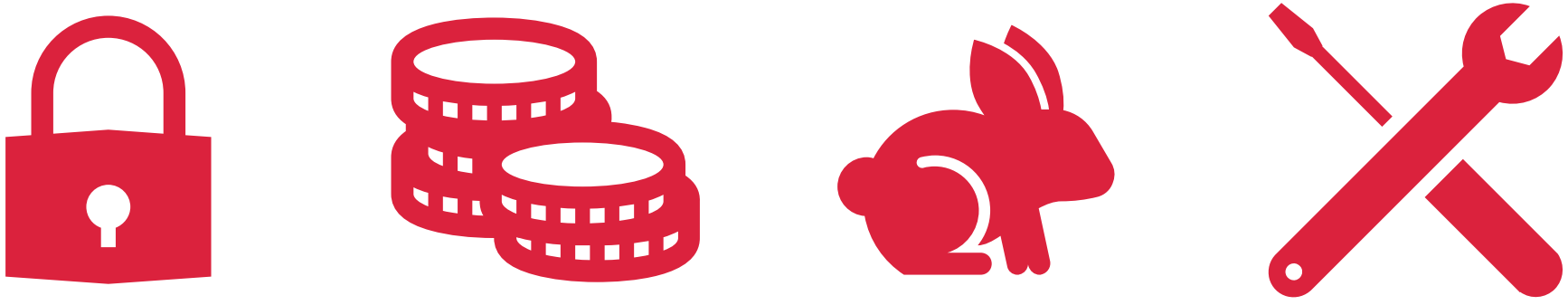
/ Integrated Command and Control System





/ Integrated Command and Control System

- Standard Communication Interface – Command and Control (SCI-CC) via AMQP
- All operator-side equipment is COTS
- Nationwide network of AMQP brokers
- Cross-compatibility between manufacturers
- All systems integrated on one operator station with single sign-on
- Safety of inputs and outputs is guaranteed by procedures instead



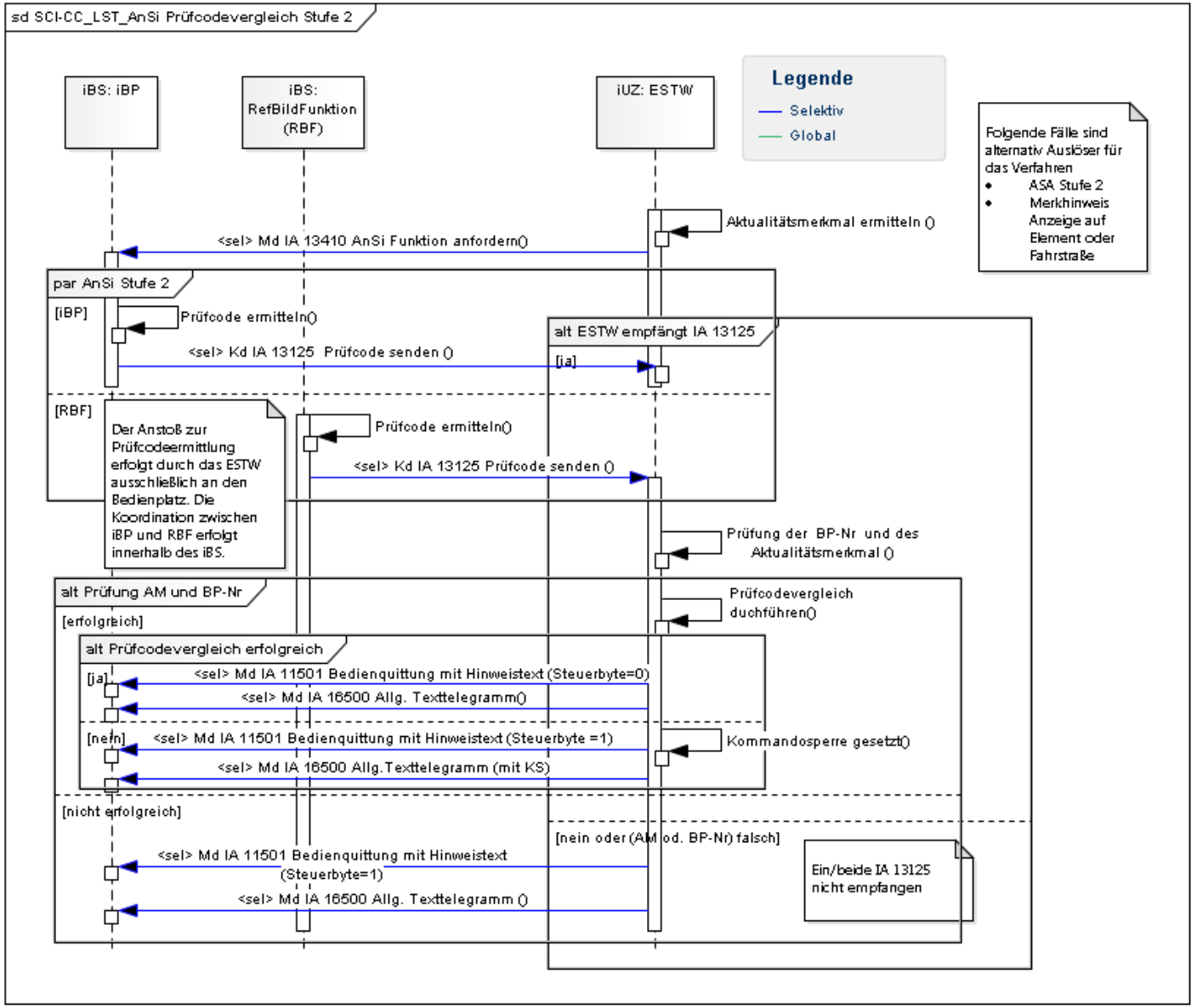
/ Development of complex systems for railways

```
private StateMachine<UserInteractionStateDbag, UserInteractionEventDbag> createStateMachine() {
    return new StateMachineBuilder<UserInteractionStateDbag, UserInteractionEventDbag>(
        UserInteractionStateDbag.GRUNDZUSTAND).onEnter(UserInteractionStateDbag.GRUNDZUSTAND, () -> {})
            // Self
            .addTransition(UserInteractionStateDbag.GRUNDZUSTAND, UserInteractionEventDbag.MTL_D,
                this::isFreeAreaEvent, UserInteractionStateDbag.GRUNDZUSTAND, () -> {
                    LOG.info("Ignoring left mouse click at free area."); //$NON-NLS-1$
                }) //
            .addTransition(UserInteractionStateDbag.GRUNDZUSTAND, UserInteractionEventDbag.MTL_D, () -> {
                    return !isShuntingRouteStart() && !isTrainrouteStart() && isUiElementEvent();
                }, UserInteractionStateDbag.GRUNDZUSTAND, () -> {
                    LOG.info("Ignoring left mouse click at uiElement {}.", eventUiElement.getId()); //$NON-NLS-1$
                }) //
            .addTransition(UserInteractionStateDbag.GRUNDZUSTAND, UserInteractionEventDbag.MTR_D, //
                this::isUiElementEventInsensitive, UserInteractionStateDbag.GRUNDZUSTAND, () -> {
                    LOG.info("Ignoring right mouse click at uiElement {}.", eventUiElement.getId()); //$NON-NLS-1$
            }) //
            // DIB_01
            .addTransition(UserInteractionStateDbag.GRUNDZUSTAND, UserInteractionEventDbag.MTL_D,
                this::isTrainrouteStart, UserInteractionStateDbag.FS_START_SELEKTIERT, () -> {
                    startRouteSelection(FahrstrassenTyp.ZUG, VQ_TRAINROUTE_START_SELECTED);
            }) //
            // DIB_02
```

/ What are our requirements from the customer?

<table>
<tr>
<td>415.906
4-87</td>
<td>Muss</td>
<td>Telegrammtabelle IA 13000
<table>
<tr><th>Byte-Nr</th><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th></tr>
<tr><td>-</td><td colspan="8">Informationsart 13000 (32C8H)</td></tr>
<tr><td>H</td><td colspan="8">Bedienplatznummer</td></tr>
<tr><td>H+8</td><td colspan="8">Aktualitätsmerkmal</td></tr>
<tr><td>H+14</td><td colspan="8">ErgBedMit (Ergebnis Bedienermitwirkung)</td></tr>
<tr><td>H+15</td><td colspan="8">PrüfCodeBP</td></tr>
<tr><td>H+19</td><td colspan="8">Transaktionsnummer (Bedienung)</td></tr>
<tr><td>H+20</td><td colspan="8">Reserve (nicht definiert)</td></tr>
<tr><td>H+25</td><td colspan="8">Sicherungsanhang</td></tr>
</table>
</td>
<td>iUZ:ESTW
iUZ:ETCS-Z
iBS:iBP</td>
<td>-</td>
<td>CR2292</td>
</tr>
</table>

/ What are our requirements from the customer?

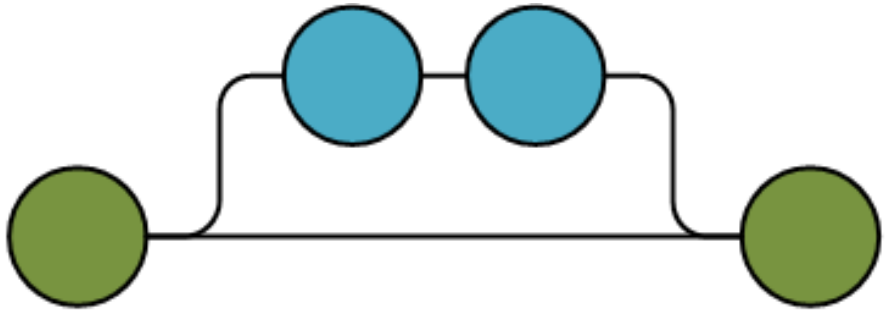


/ How do we adapt the requirements?

- Use generative tools to create common libraries for communication data structures
- Generating code from sequence diagrams not really a thing
  - (As far as I know)
- One sequence diagram usually contains more than one system that is part of the communication
  - Not all of these systems are developed by us
  - Not all of the systems we develop are developed by the same dev team
- But: The sequence diagrams „hide“ one or more state diagrams that we care about

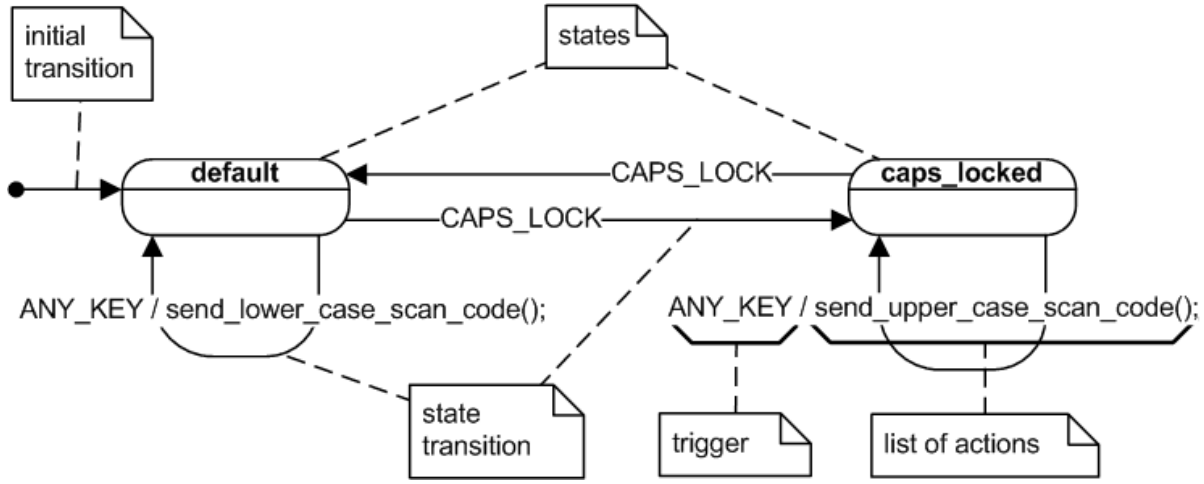
Find a state modelling tool suitable for our usage in railway applications

/ Suitable?



- Work together with commonly used version control systems
  - Text based file format
  - Limited complexity in storage format

/ Suitable?



- Some form of visualization for the state machine
  - Bonus points if the visualization actually has some meaning for non-developers (like domain experts)

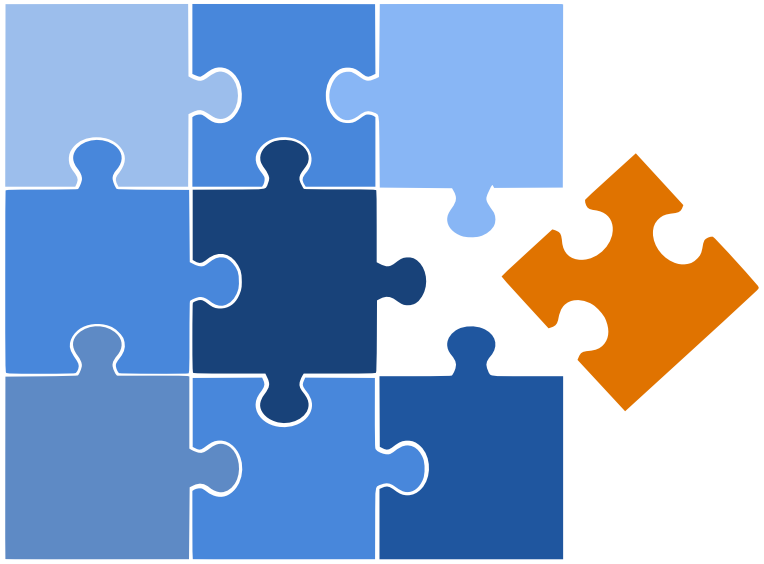
/ Suitable?

- Support for multiple target languages
  - Java
  - C++
  - Python (bonus)

/ Suitable?

- Usable for software developers used to classic, imperative programming
  - Synchronous languages can be a bit intimidating at times
  - I don’t want to teach a 3-week course on special languages for every new dev

/ Suitable?



- Adaptable/Integratable in existing systems
  - Support for various existing systems and toolchains

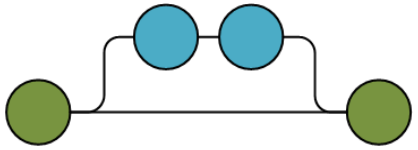
/ Suitable?

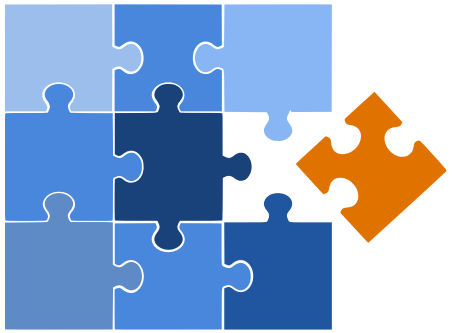
- Usable in certified safety-critical systems
  - Either complete certified language/semantics/compilation
  - Or „clean“ generated code, that can be certified itself

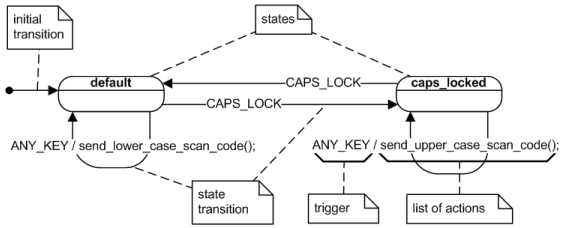
/ Suitable?

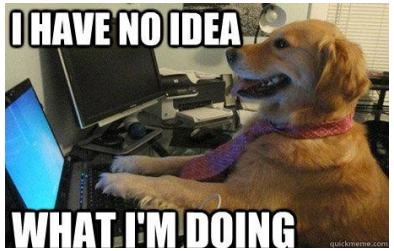
- Long-term availability and usability
  - Our systems need to be supported for multiple decades
  - Common support time is 20-30 years, on ocasions up to 40 years

/ Suitable?



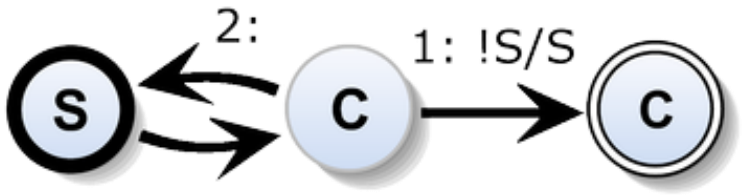






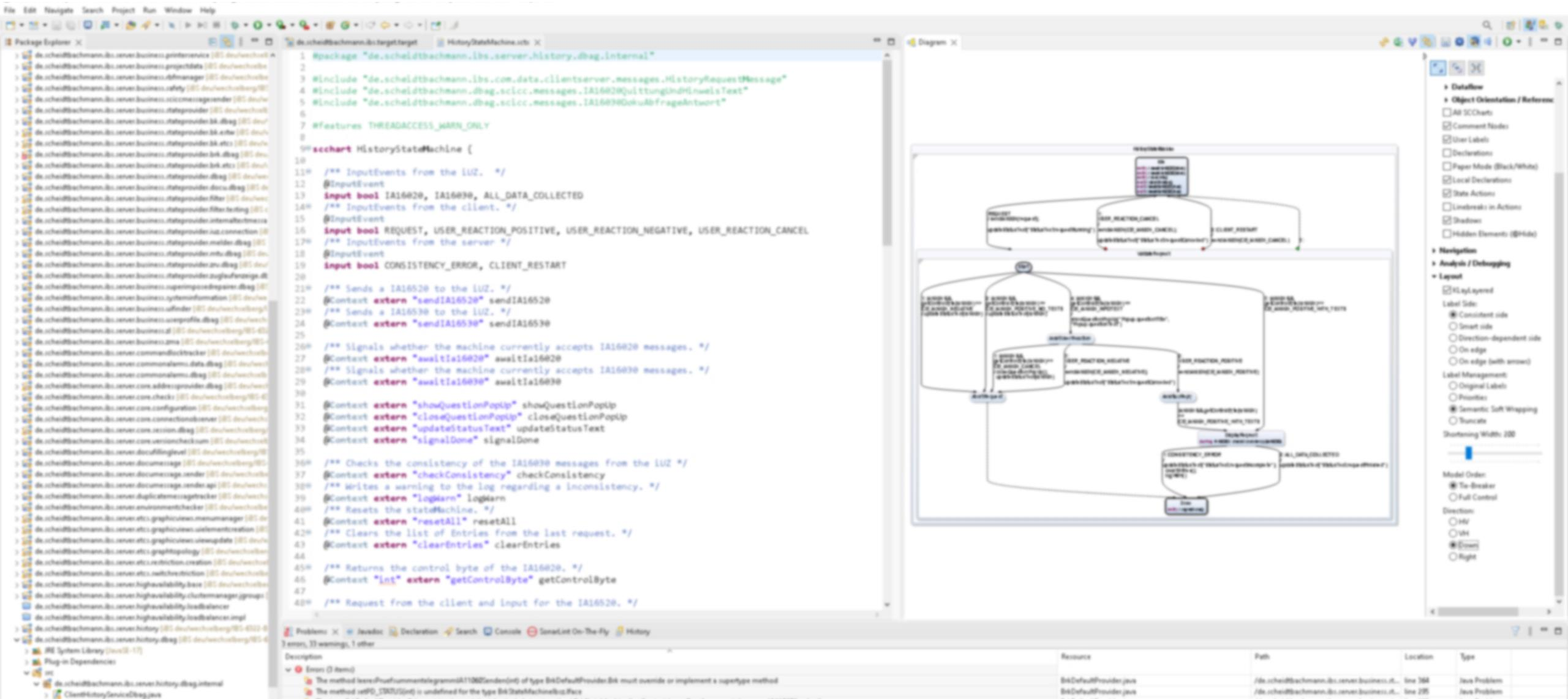
- Given all these constraints there are two options
  - Exceptionally well adaptable, commercial solution
  - Free and open source tool that we can adapt (and/or keep running) ourselves

/ (Mostly) Suitable!



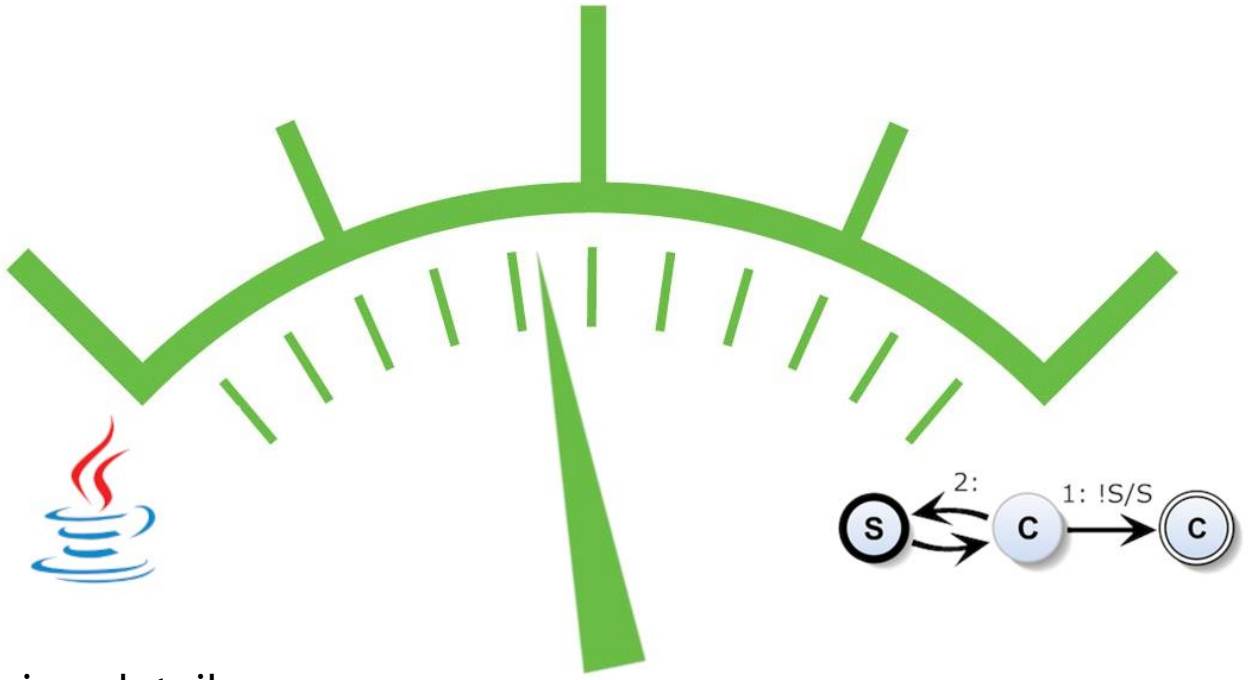
- Sequential constructiveness reduces needed training for new devs
- Existing code generation for Java and C
  - Is the code up-to-spec to be used in safety-critical systems?
- Flexible execution model
- Plain text DSL easy to read
  - Caveat: Getting all the details of the grammar right needs some time
- Flexible transient visualizations

/ Embedding SCCharts in our development



/ Task 1: Deciding on a separation of concerns

- What parts should be handled by SCCharts, what parts by the host language?

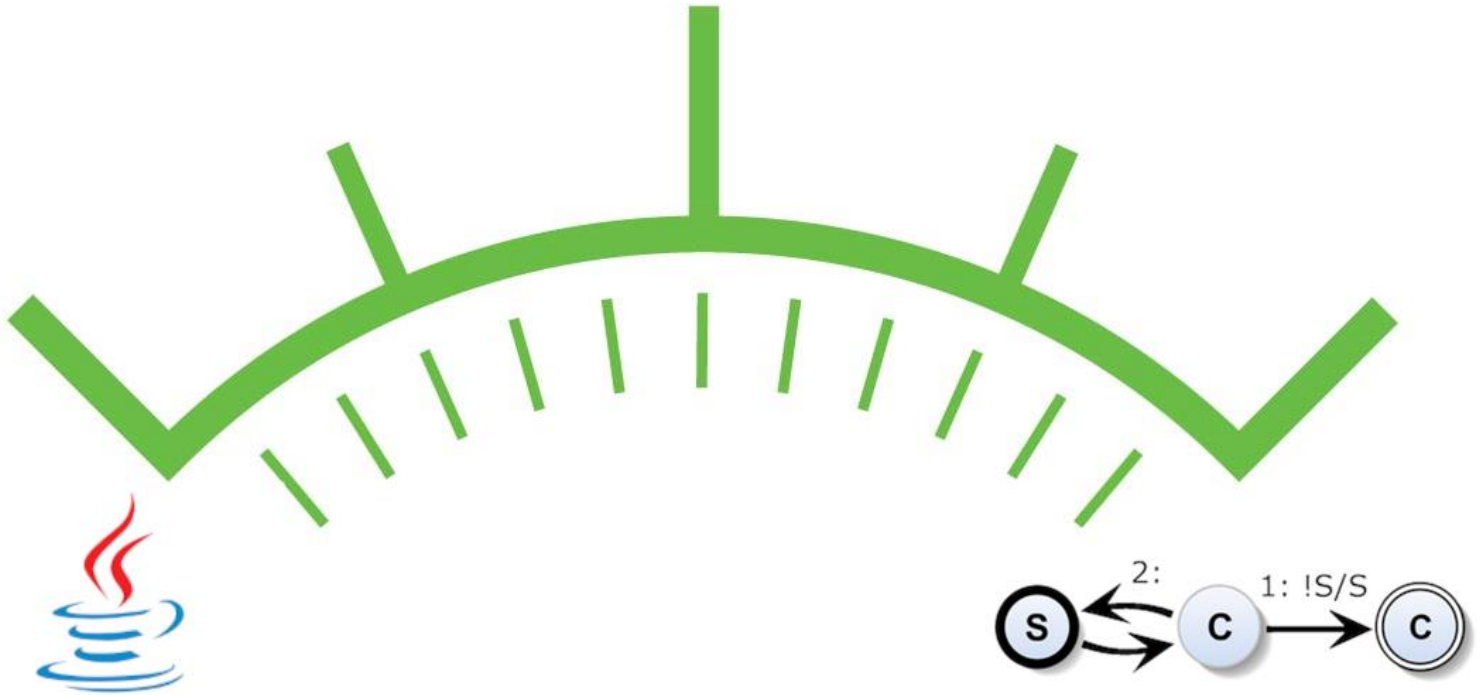


Threading
Messaging
All the annoying details

Coordination
Flow-Control
Business-Logic

/ Subtask 1: Main Event-Loop

- Who should be responsible for the main application control?



/ Task 2: Scheduling the execution of SCCharts

- Free-running mode
  - Run the SCCharts tick loop as fast as it wants

- „Fixed“ time mode
  - Have a (semi-)fixed scheduling for the tick loop

- Event-based mode
  - Only run the tick loop when there is some actual work to do

/ Subtask 2: Handling concurrency surrounding SCCharts

- The modelled code is thread-safe, regarding concurrency/concurrent events
- However, the generated code is not entirely thread-safe
  - Running the tick() function from multiple concurrent threads causes bad things
- Our application is highly concurrent, so we need to address that
- Solution: Every SCCharts objects carries a dedicated execution thread that is triggered with the current events
- Caveat: Every access to inputs or outputs of the SCChart needs to be performed in this thread

/ Task 3: Choosing a compilation chain

- SCCharts provides multiple ways to compile the model to code

- Netlist-based
  - Thoroughly tested
  - Full support for all advanced features
  - But: Code not human-readable / Unusable for certification

- Priority-based
  - Not always stable
  - Unclear feature support (though I guess it should be feature-complete)
  - No Java target (as far as I know)

/ Task 3: Choosing a compilation chain

- We might accept a reduction in the feature set of the language if the generated code is „clean“ and human-readable

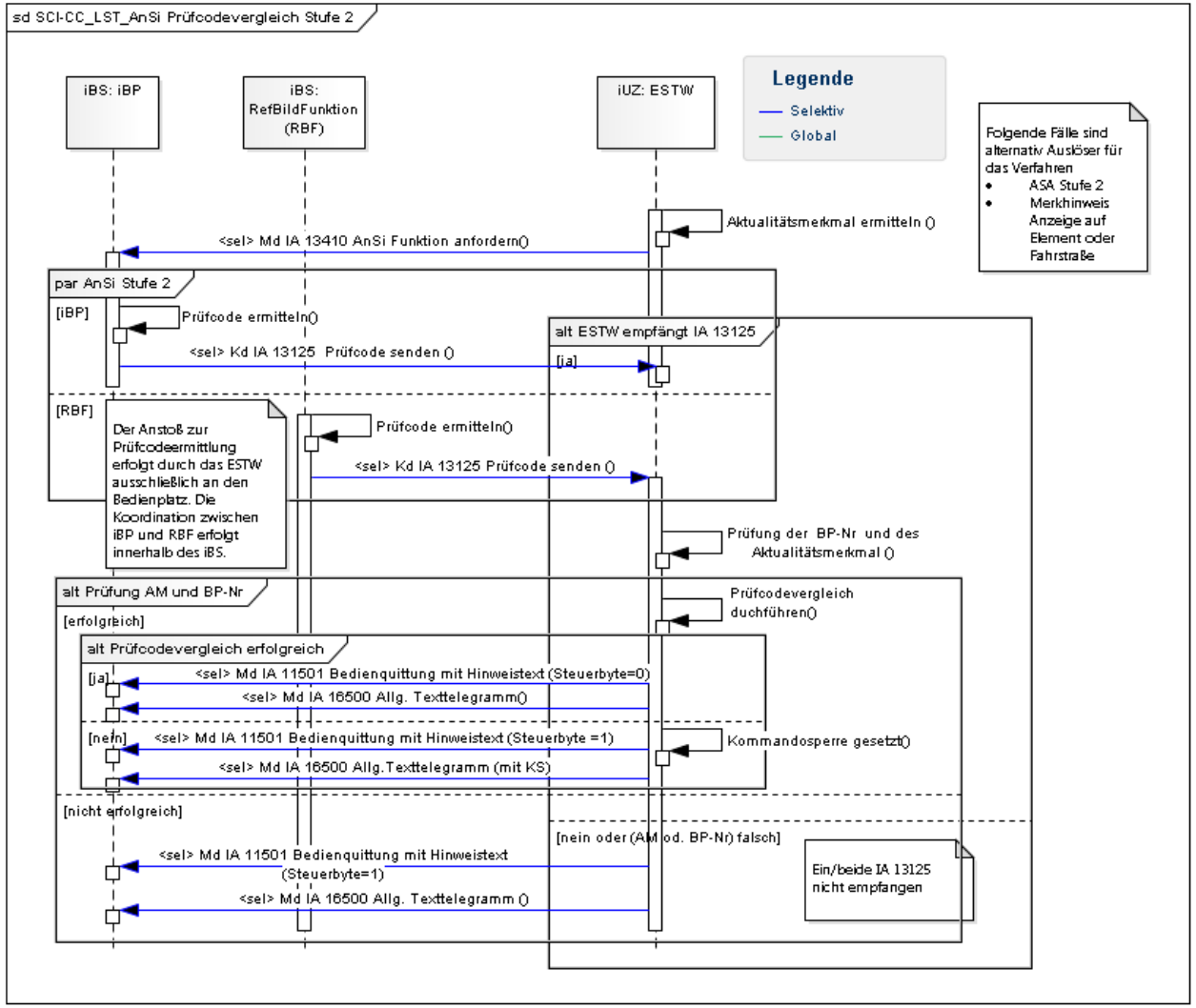
- „New“ compilation chain: „Lean“ State-based
  - Reduced feature set (especially in regards to dependent concurrent regions)
  - Mostly direct mapping of model to generated code possible
  - Generated code is WAY more verbose, but mostly human readable

/ Future Work / Options

/ Future Work – Integrating multiple models

- Remember me saying:

  The sequence diagrams „hide“ one **or more** state diagrams that we care about
- These SCCharts are usally deployed in different target systems
- But: Could we maybe do some analysis whether these systems behave, if we know something about the communication?



/ Future Work – Model Checking

- Can we check the SCCharts against certain safety and/or liveness criteria?

- Currently ongoing research @RTSYS Group

www.scheidt-bachmann-st.de

SCHEIDT&BACHMANN