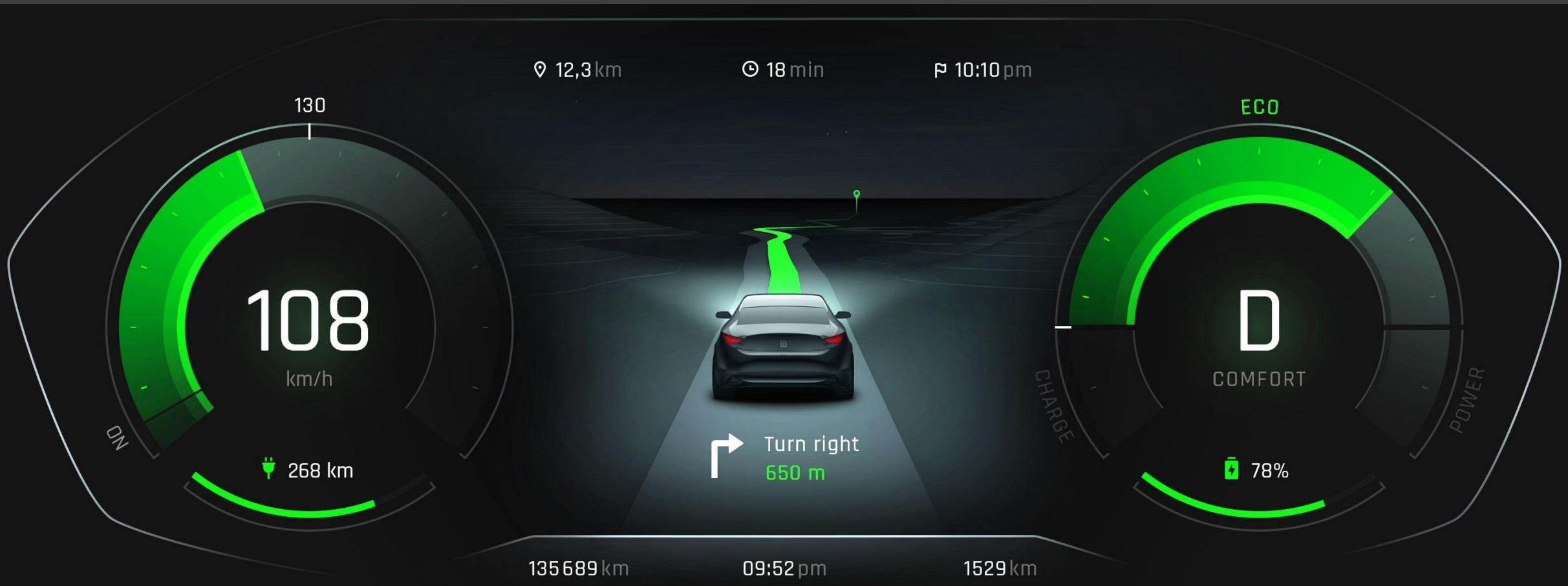
Case study: Cluster HMI - from prototype to production-ready

Bergmann-Eggers, Sören

July 2019

Elektrobit



About Elektrobit (EB)

Technical competencies

EB's technical core competencies are development of automotive-grade (software) products and engineering services.

Global presence

Development and business offices in Austria, China, Finland, France, Germany, India, Israel, Japan, Romania, and USA.

Employees

More than 2200 employees* worldwide. Spans three continents and ten countries.

Continental AG

Wholly owned, independent subsidiary of Continental AG.

Consistent growth

Average growth (CAGR) > 10 %

100+ million

Over 100 million vehicles on the road and one billion embedded devices.

*May 2018, incl. Argus, excl. e.solutions.

Driving technology further — products & solutions

Automated driving

- Hardware and software products for development, test, visualization, and validation.
- Key software components to bring automated driving functions and systems to serial production.

Vehicle infrastructure

- AUTOSAR standard
- Single- & multi-core OS
- Functional safety OS
- Embedded security
- Automotive networks, e.g. Ethernet

Connected car

- Intelligent big data analytics & online diagnostics
- Scalable backend infrastructures
- Cyber security solutions plus modular add-ons by Argus
- Software updates over the air

User experience

- Navigation client for connected use cases
- Electronic horizon provider enabling map-based ADAS functions
- Model-based development of multimodal user interfaces
- Augmented reality solutions

Consulting services

- Consulting services for functional safety and software architectures
- Lean software development
- Established agile processes

Verification and validation

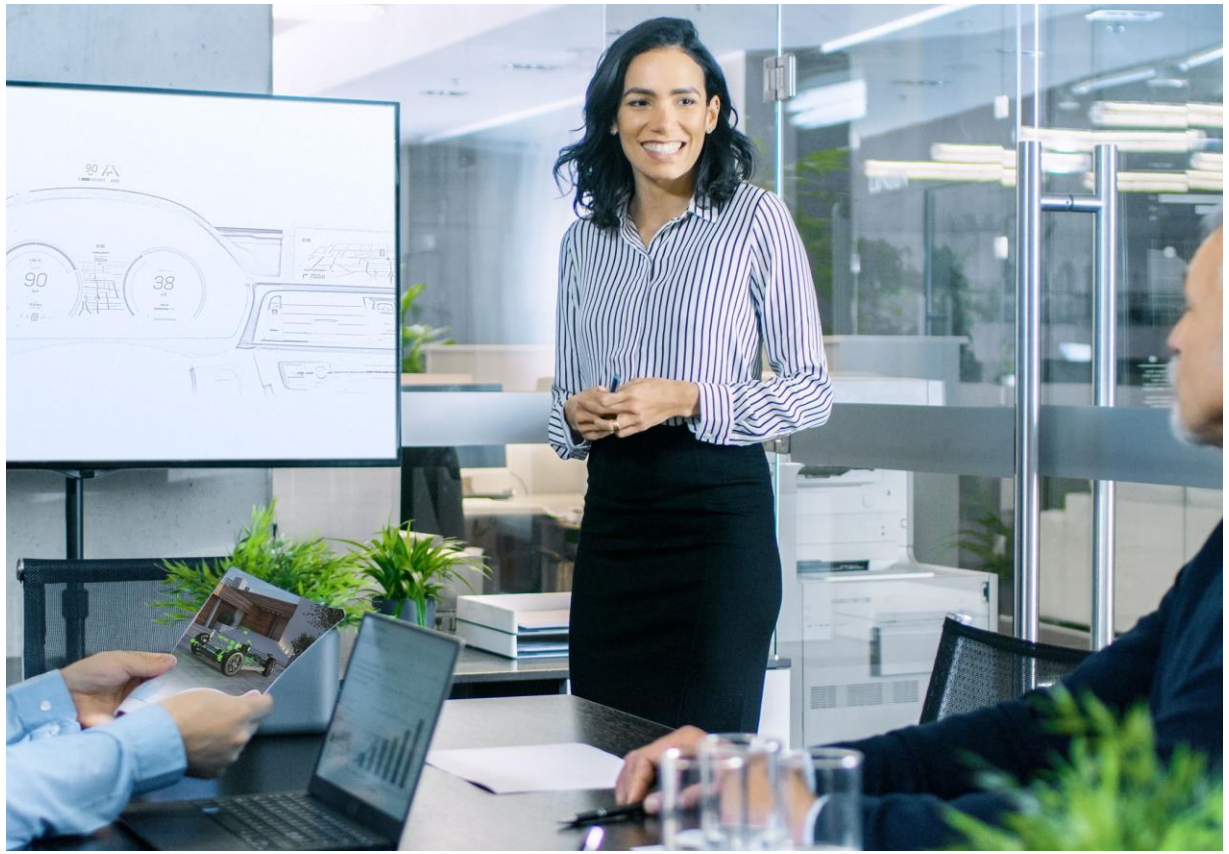
- End-to-end testing of complex embedded software systems
- Test concept development
- Independent verification and validation of software systems

The prototype

Cluster instrument HMI project kick-off

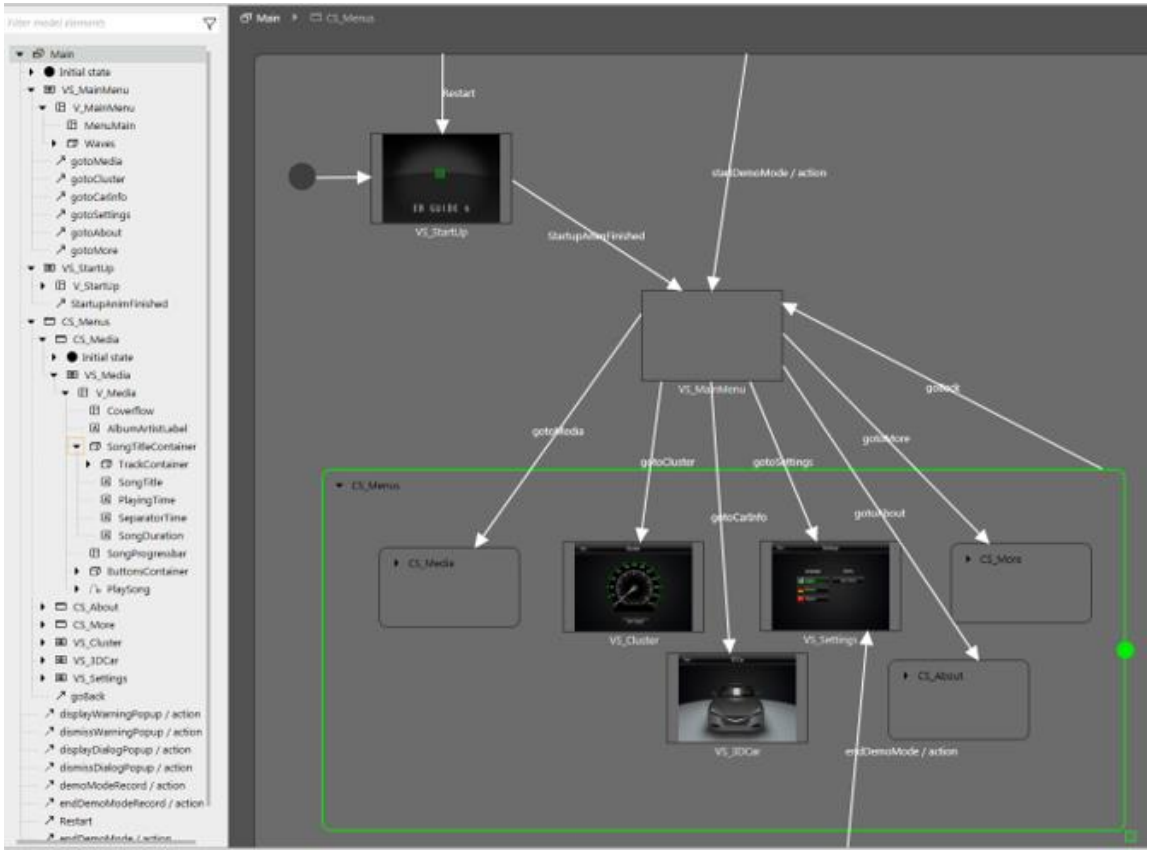
Request for a cluster HMI

- Team is requested to create a prototype for a cluster HMI.
- Customer already provided basic design prototype.
- Prototype is needed fast.
- Target hardware is roughly specified.
- Development should start immediately after the prototype has been finished.



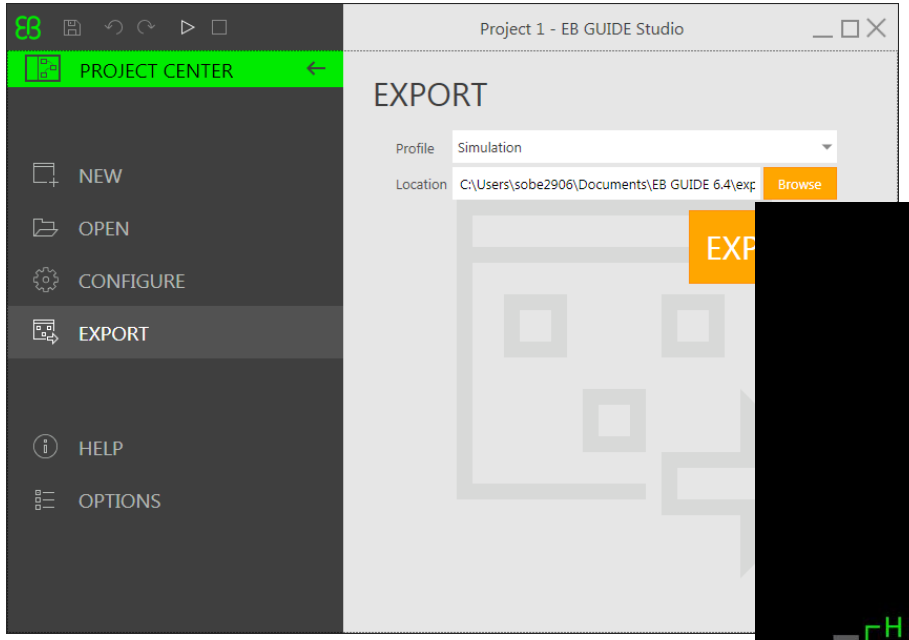
First implementation of prototype with EB GUIDE

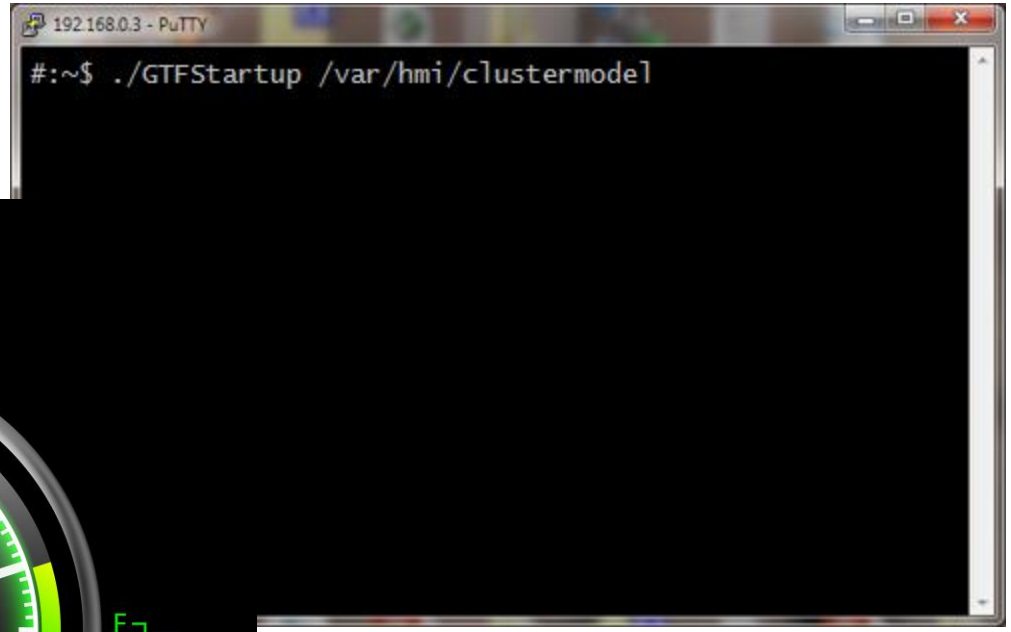
- Implementation on desktop PC
- Simple state machine with a couple of views
- Views based on design prototype by hand
  (or .psd import by EB GUIDE)
- Data is hard-coded in data pool, can be manipulated in EB GUIDE Monitor.
- Basic design demo without data providers is running on PC after couple of hours.

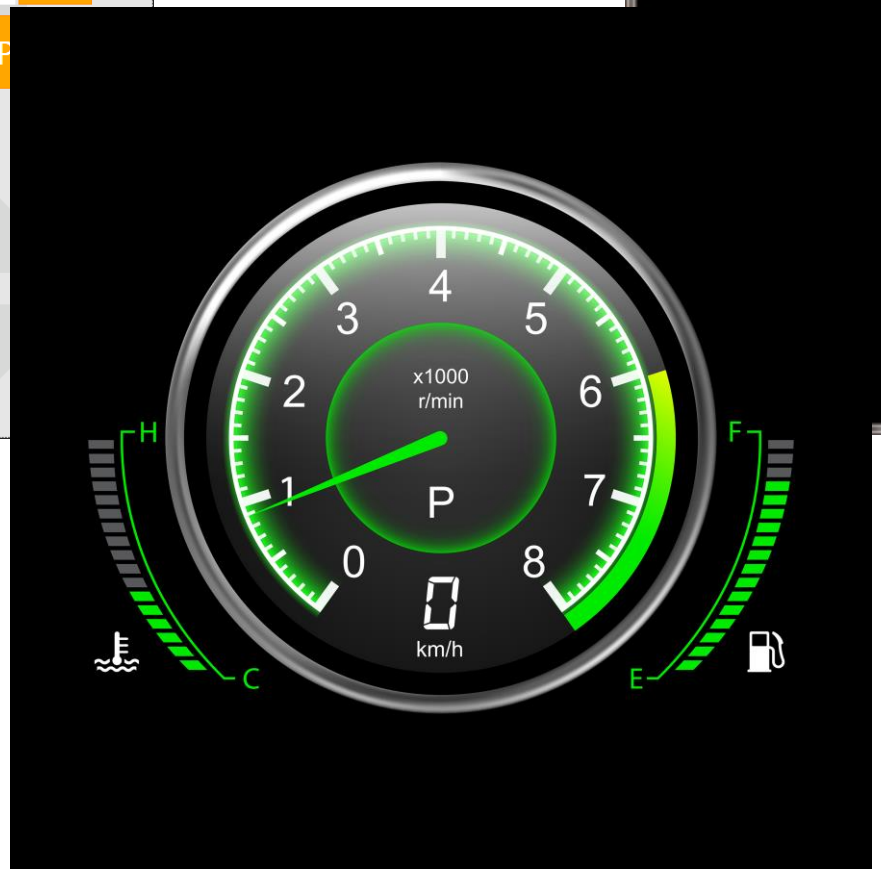


Elektrobit

Deployment to target

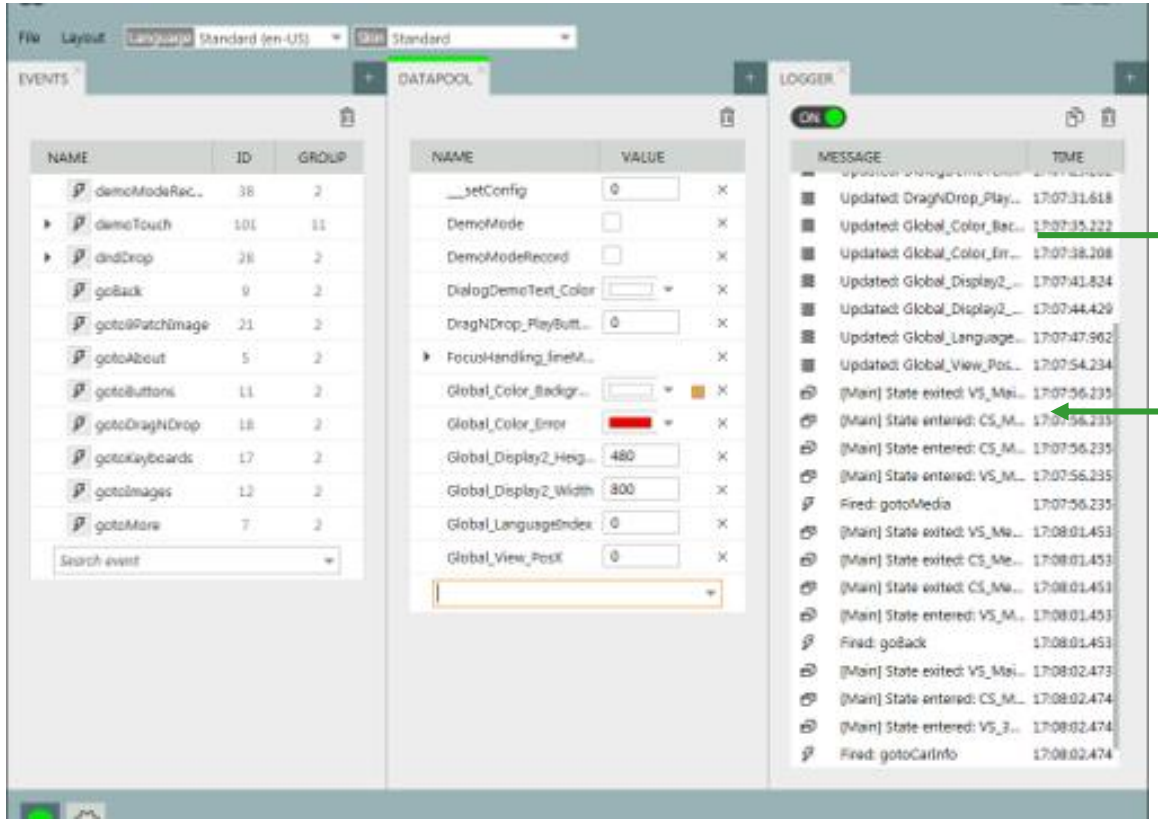






First tests on the target

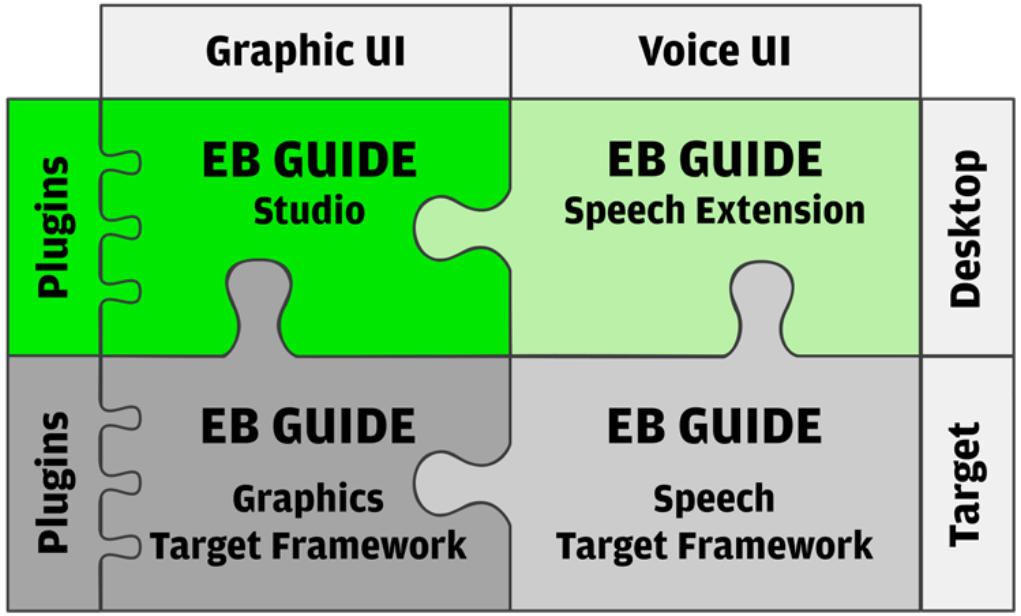
Connect EB GUIDE Monitor



Manipulate data pool, send events

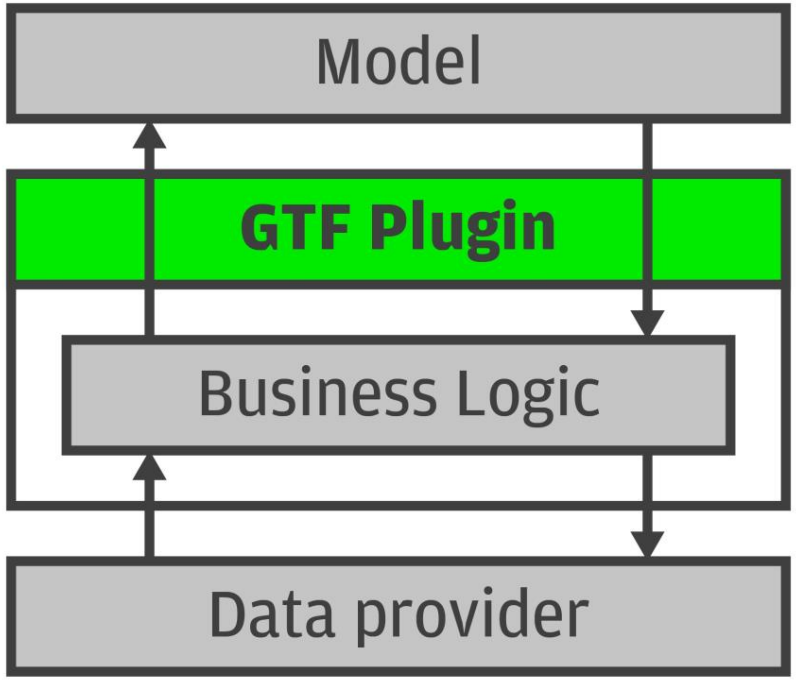
The EB GUIDE Graphics Target Framework (GTF)

- The GTF is an executable ported to the target platform.
- GTF loads exported model data on startup.
- Allows running the same model export on desktop and any target platform.
- Extendable by plugins



Processing data

- For processing data a GTF plugin is created.
- Interface to model data pool and event system is generated by EB GUIDE.
- First implementation sends events and value updates to model.
- Connection to a data provider is implemented (e.g. TCP or CAN).
- In first implementation everything is in one component.
- Connection works → Feeling of “almost done” (even if probably inappropriate).



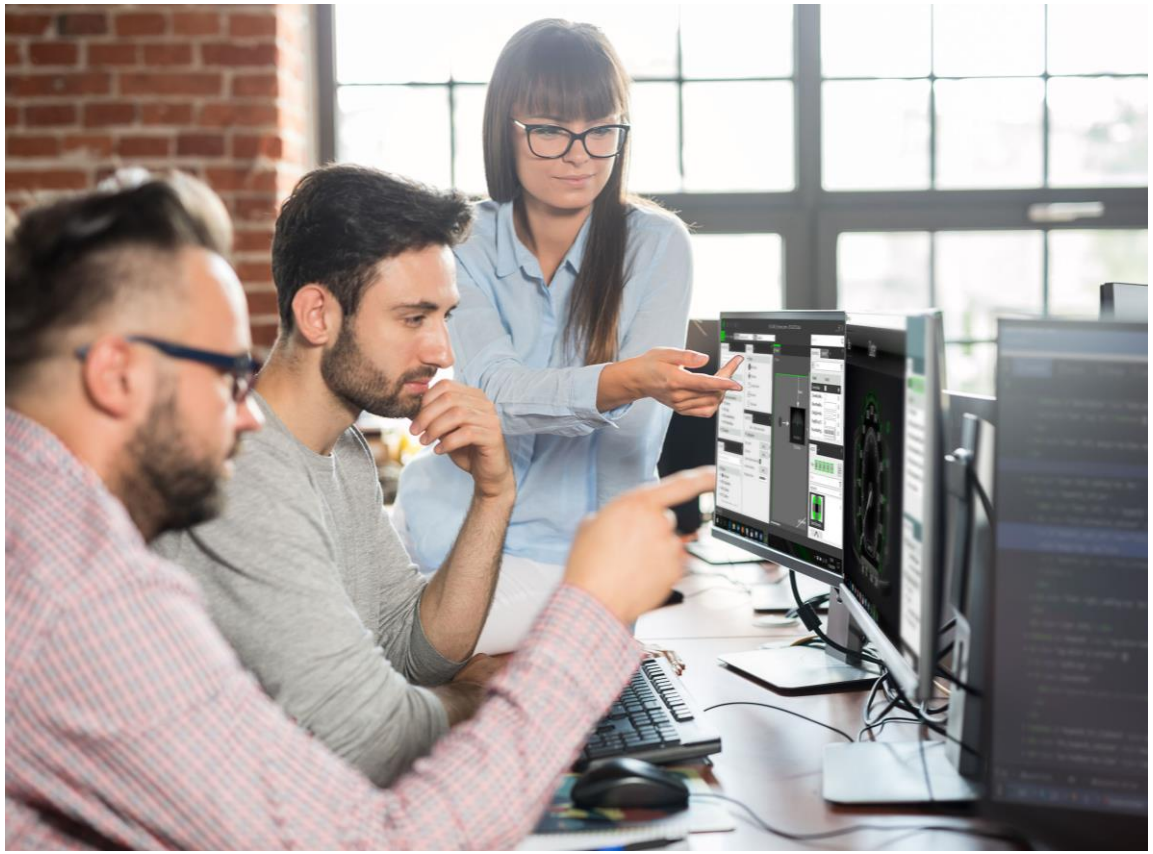
The feature set grows - so does the team

Growing team of EB GUIDE users

Multi-user support

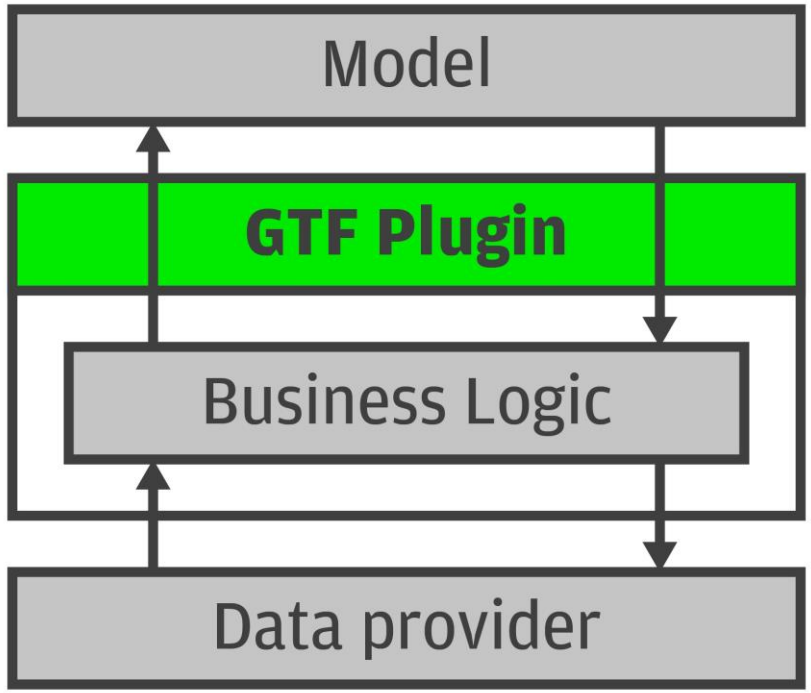
- EB GUIDE components stored in individual files.
- EB GUIDE file format is human-readable json code.

- **Benefits:**
  - Multiple users can work on the same model.
  - File format is compatible to any code versioning system.
  - Users can merge conflicting edits in case.
  - Change history can be traced in diffs.



Prototype reaches borders

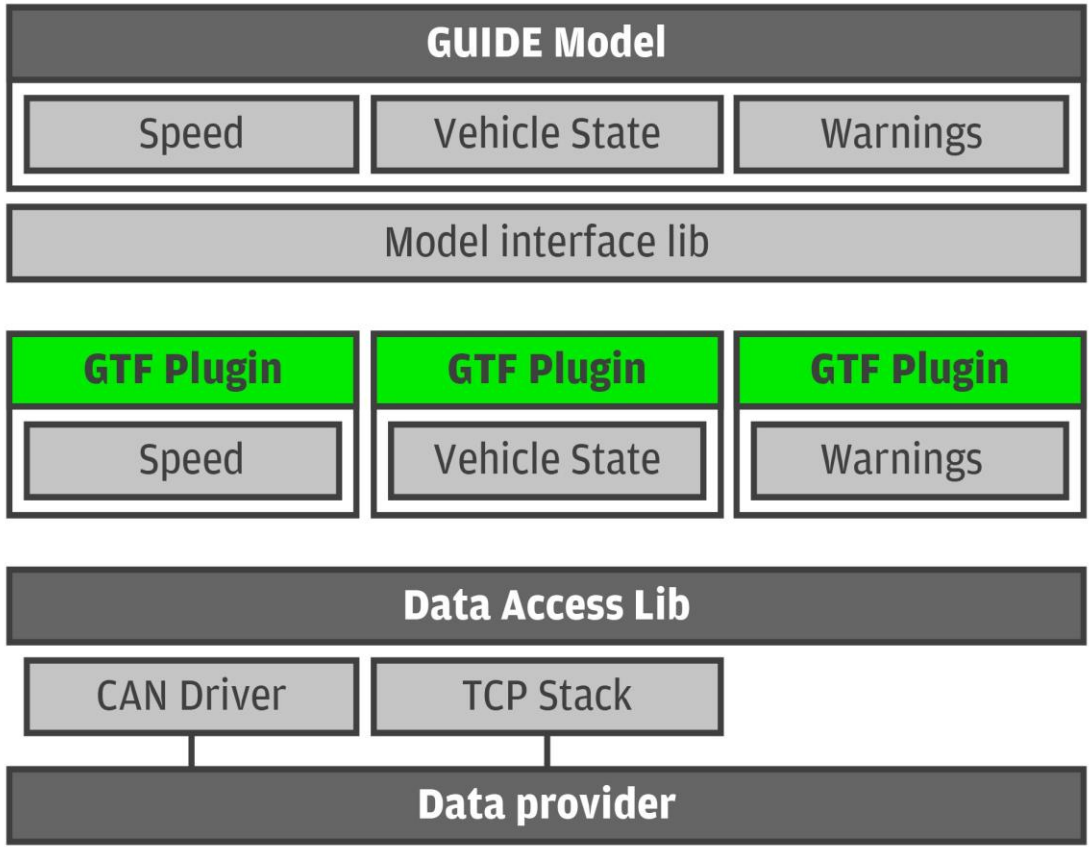
- Situation with prototype:
  - Business logic was designed for single technical scope.
  - Adding functions to business logic mixes up technical scopes.
  - Data arrives by different channels.
  - Code for updating model data is repetitive.
  - Business logic handles all type of channel and model updates.
  - A second component would have to repeat this implementation.



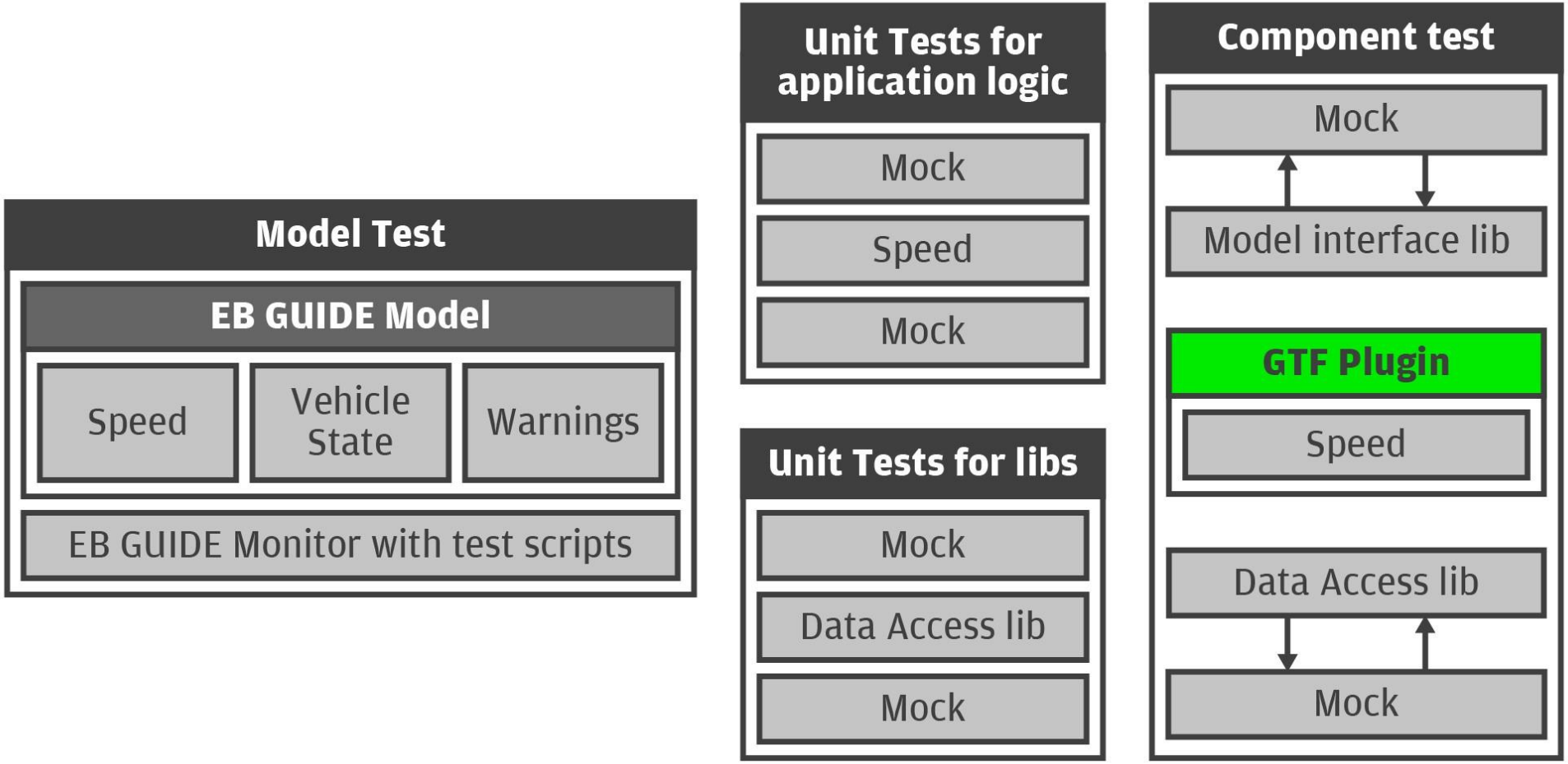
Growing team of plugin developers

- Introduce separation of concerns
  - Layer for model communication
  - Layer for data provider
  - Business logic just translates incoming data to model data and vice versa
  - Separating business logic in functional independent blocks

- Benefits:
  - Application logic is platform-agnostic (even GTF)
  - Independent testability of logic components and access libraries
  - Automated tests for model design (EB GUIDE monitor scripts)
  - Test scope can be defined layer by layer.

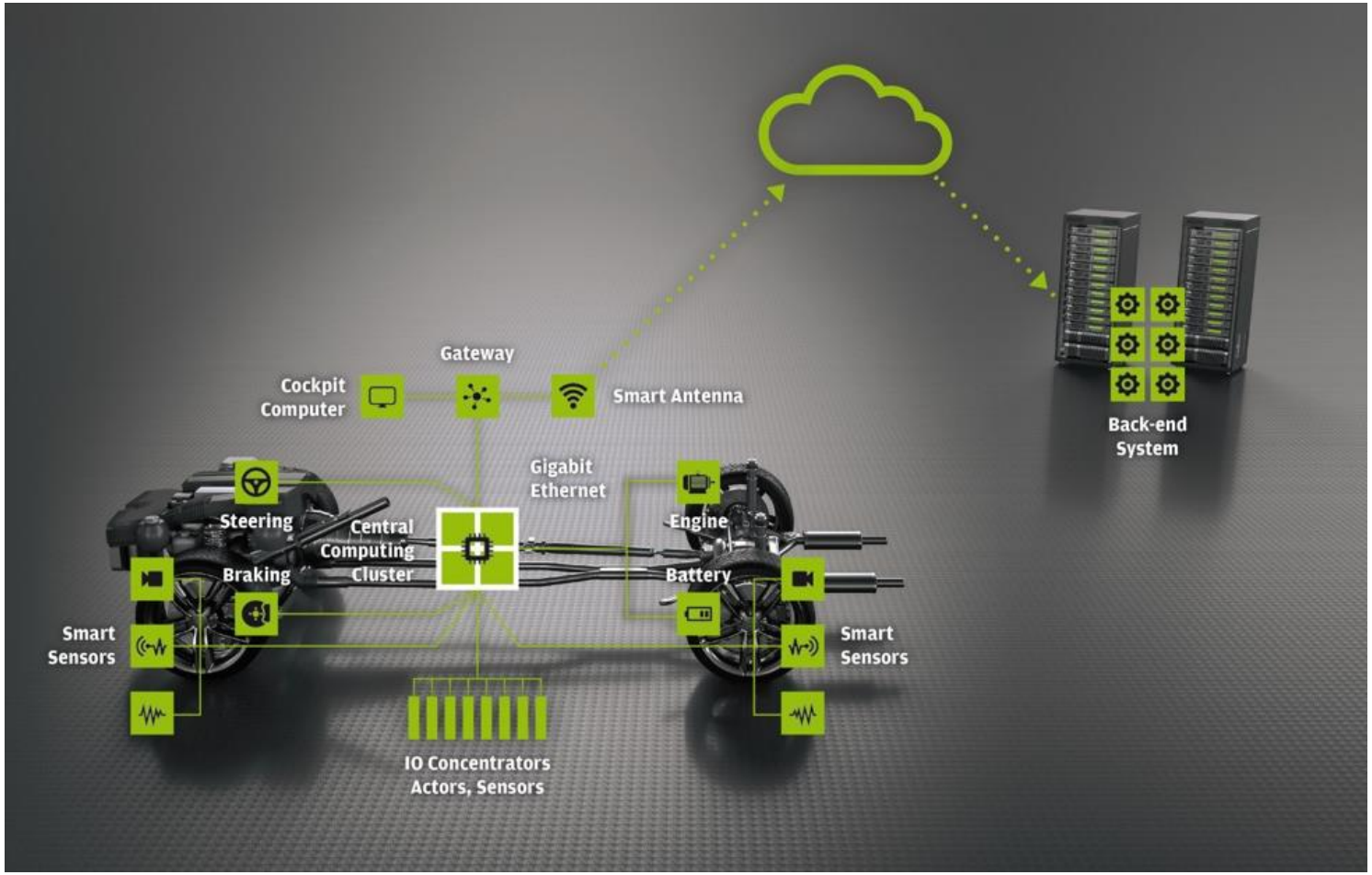


Testing setups



Reality check

System integration test



System integration test

System failure

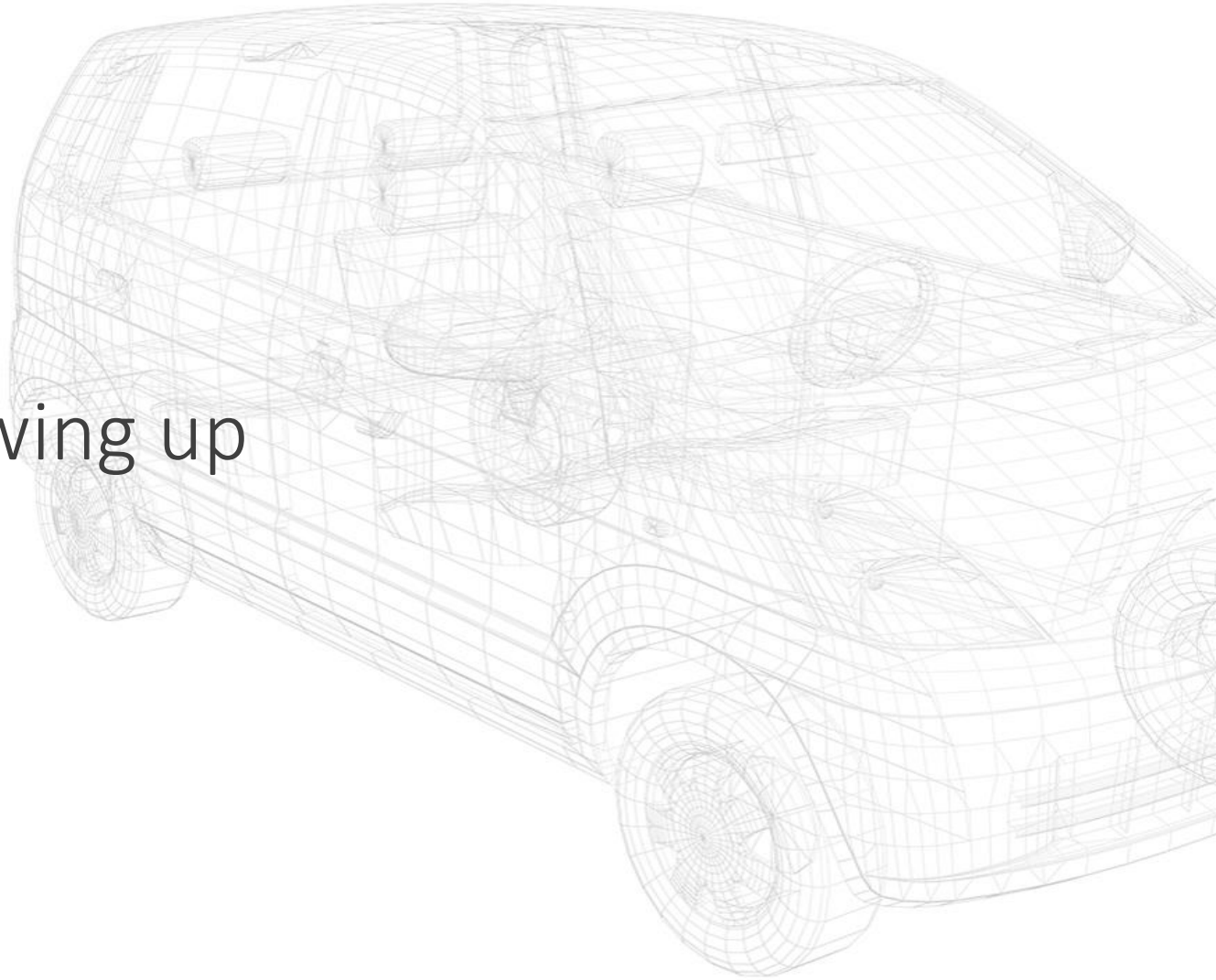
- Some ECUs deliver the expected data.
- Some don’t or do sporadically.
- Some interpreted spec differently than the HMI team.
- A crash of the HMI occurred.
- Performance of HMI is unsatisfying.

System integration test

Fast system recovery

- Issues can be analyzed and fixed quickly during the workshop:
  - Developers bring their complete toolchain to the workshop.
  - Components allow isolation of issues .
  - Separation of concern allows safe application of fixes.
  - Short build and deploy times allow direct testing of patches.
- The overall integration test is considered a success:
  - The HMI was delivered in time.
  - The HMI was started and the user experience was presented.
  - Critical issues could be addressed and solved directly during the integration test phase due to fast roundtrip times.
- Follow-up
  - Further analysis of remaining issues later
  - EB Experts help on reviewing model and source code
  - EB port team optimizes GTF to given hardware

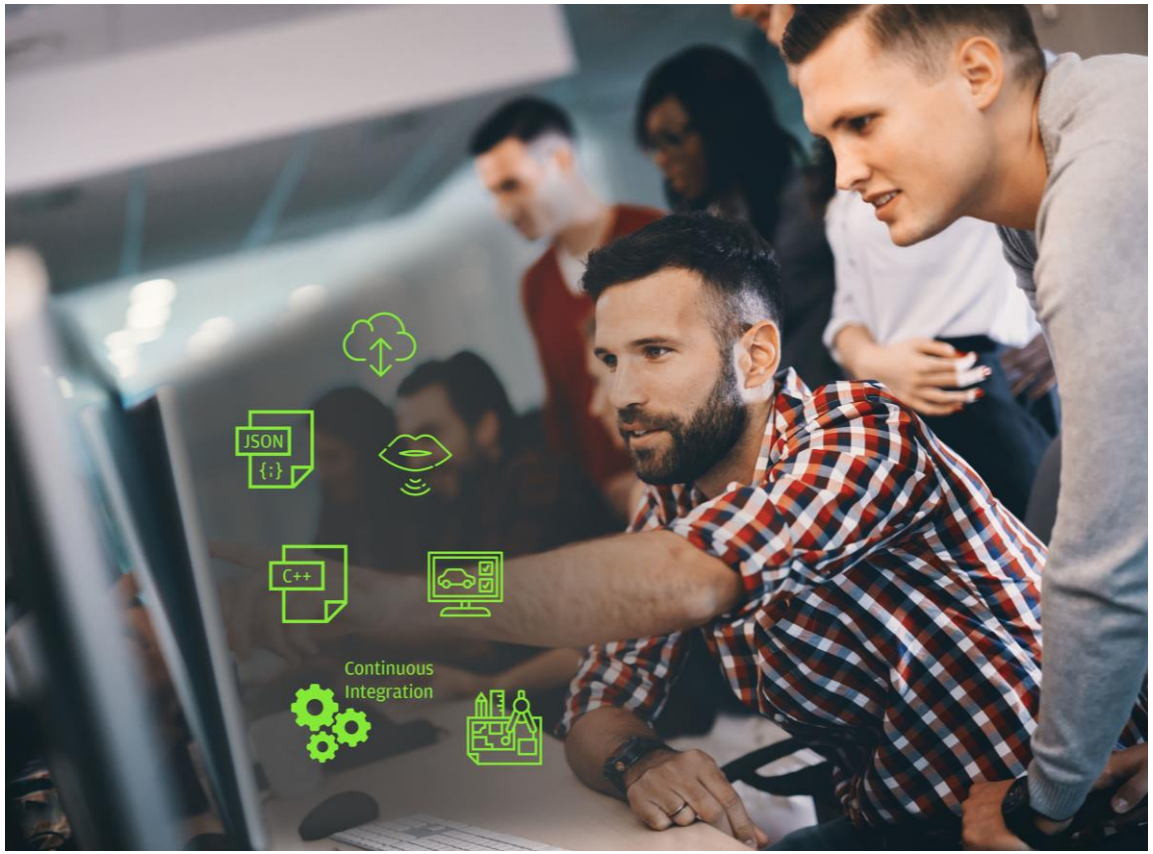
Growing up



Development has to match standard processes

- Organization has established processes for
  - Integration
  - Configuration management (git/svn)
  - Delivery

- Project structure supports these processes:
  - Libs for abstraction layers and components can be maintained separately.
  - Nightly builds can integrate all components.
  - Automated tests on different scopes can be run.
  - Model is json format (code) and plays well in the established processes.
  - Headless EB GUIDE supports continuous integration.

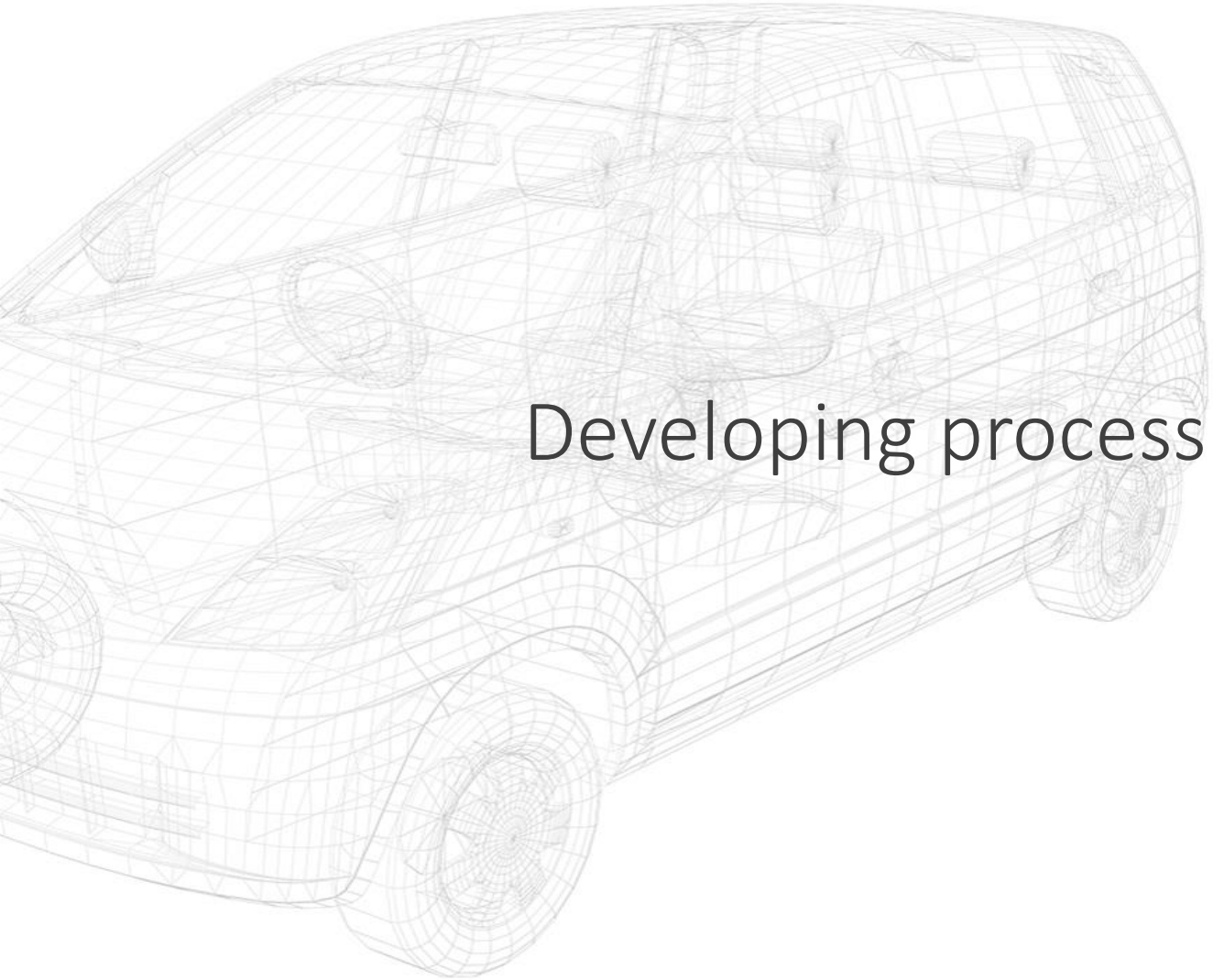


Design updates are incoming

- Design team provides frequently updates.
- Functional scope is extended, design grows, many views, views get more and more elements.
- Animations are defined.
- Integrating the design updates manually is repetitive.
- Is there a way to automate this?
  - Designers and developers agree on technical exchange format and process (e.g. psd).
  - Developers create EB GUIDE C# plugin for importing design information

Internationalization is required

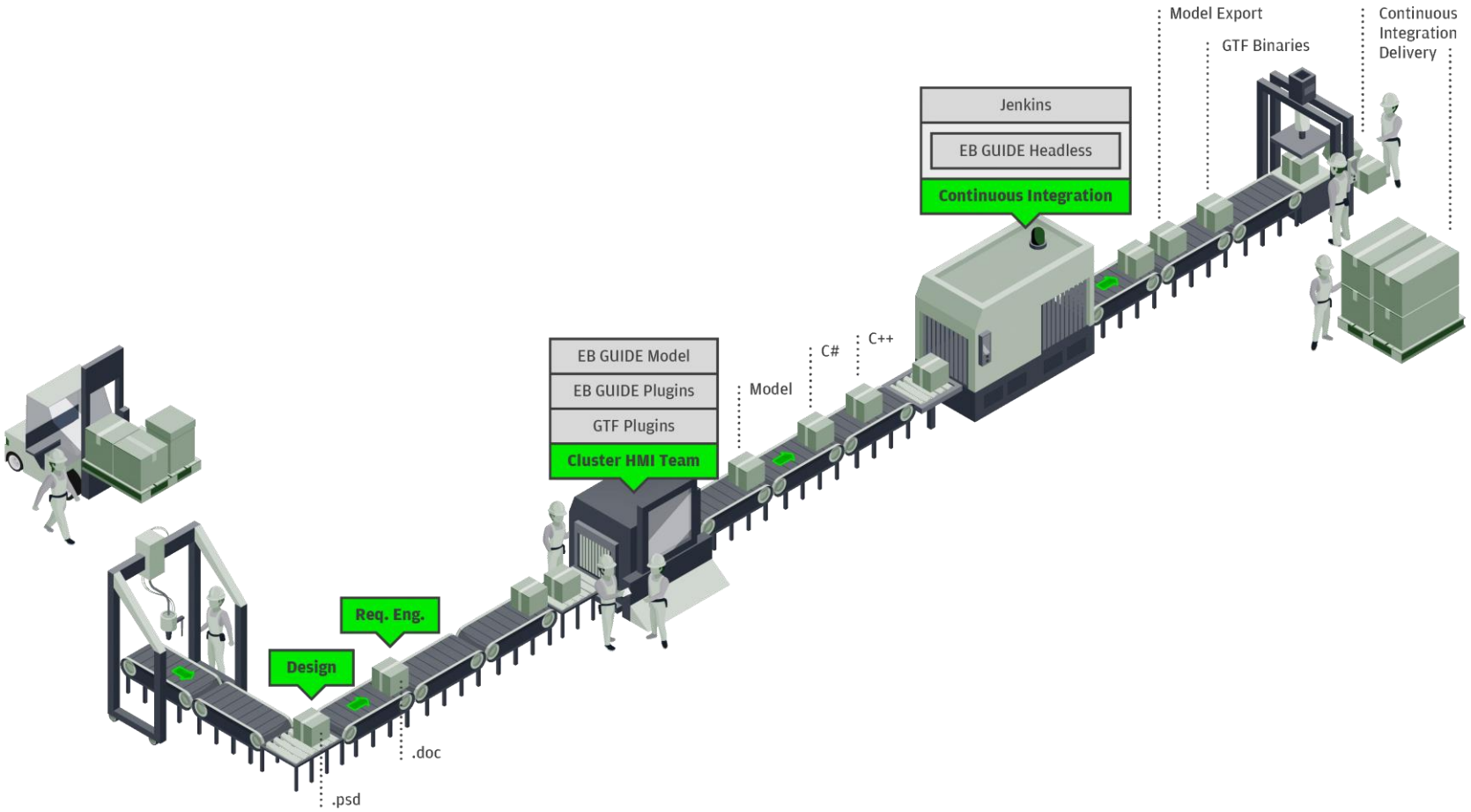
- All texts have to be translated into different languages.
- Organization has a division with given processes.
- Developer team creates EB GUIDE plugin to match these processes.
  - EB GUIDE provides basic in and export tool for texts.
  - Based on the source code the tool is adapted to processes of organization.
  - EB GUIDE plugin has access to all elements of the model and can annotate texts with meta information helping the translation team to find fitting translations.



Developing process becomes routine

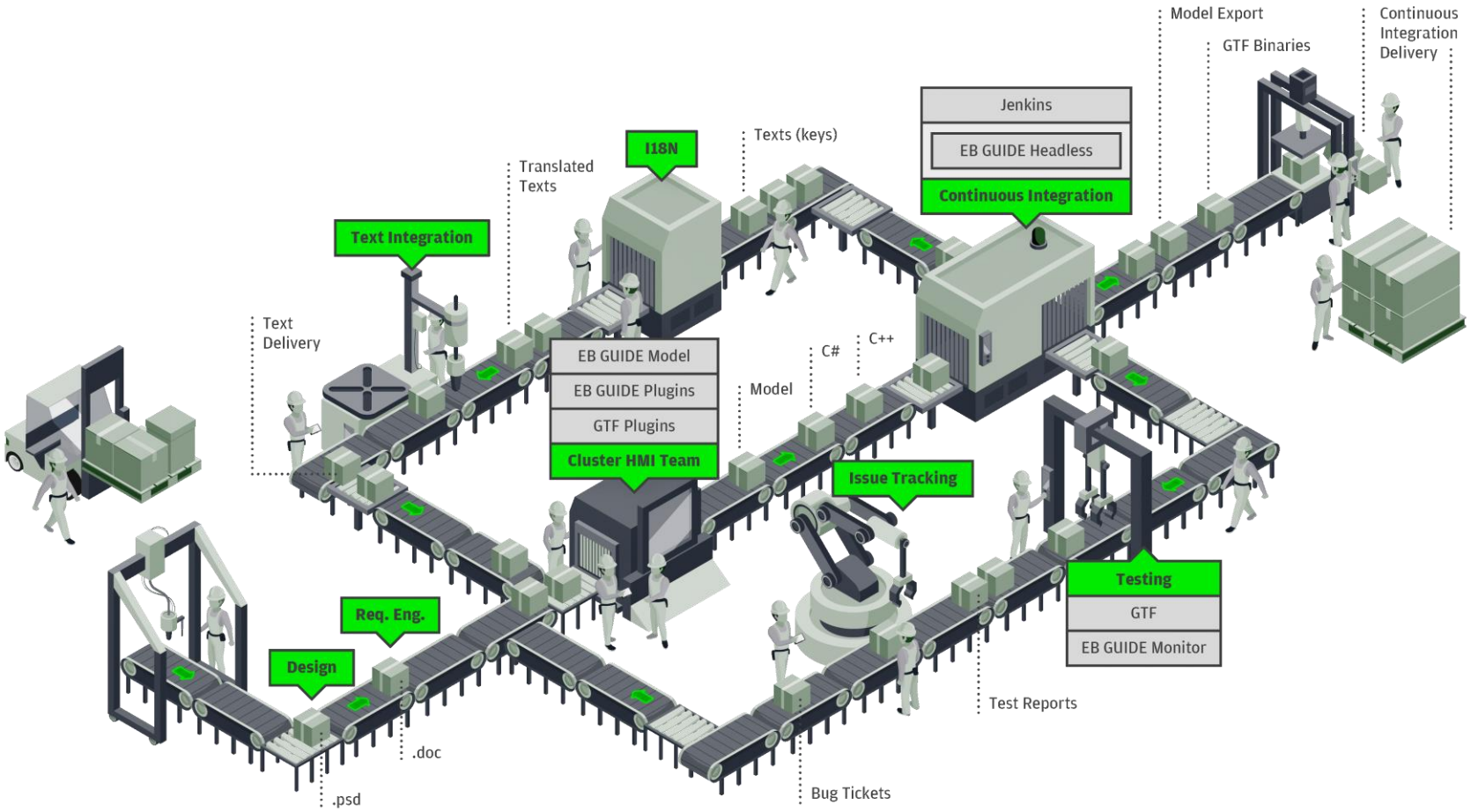
Elektrobit

EB GUIDE becomes integral part of the process workflow



Elektrobit

EB GUIDE becomes integral part of the process workflow



Get in touch!

Elektrobit

sales@elektrobit.com
https://www.elektrobit.com/ebguide