logi.cals

logi.CAD3 → logi.CLOUD

An experience report on migrating an industrial-grade IDE to the Cloud using the Eclipse ecosystem

WE EMPOWER AUTOMATION

Agenda of this Talk

logi.cals

- **Introduction (7 Minutes)**
  - Goal
  - Company focus (products, technical)
  - High-level goal
- **Experience report (20 Minutes)**
  - Architectural decisions
  - Challenges
- **Outlook (3 Minutes)**
  - Future steps
- **Q&A (5 Minutes)**

Goals of this Talk

logi.cals

- **Usage scenario of Eclipse technologies**
  - Industrial Automation
- **Sharing experiences: Make Fat Desktop App usable in Cloud Environment**
  - Overview of frameworks and projects
  - Challenges, Pitfalls
  - How we got started

About
logi.cals

logi.cals

1987
FOUNDED

35
EMPLOYEES

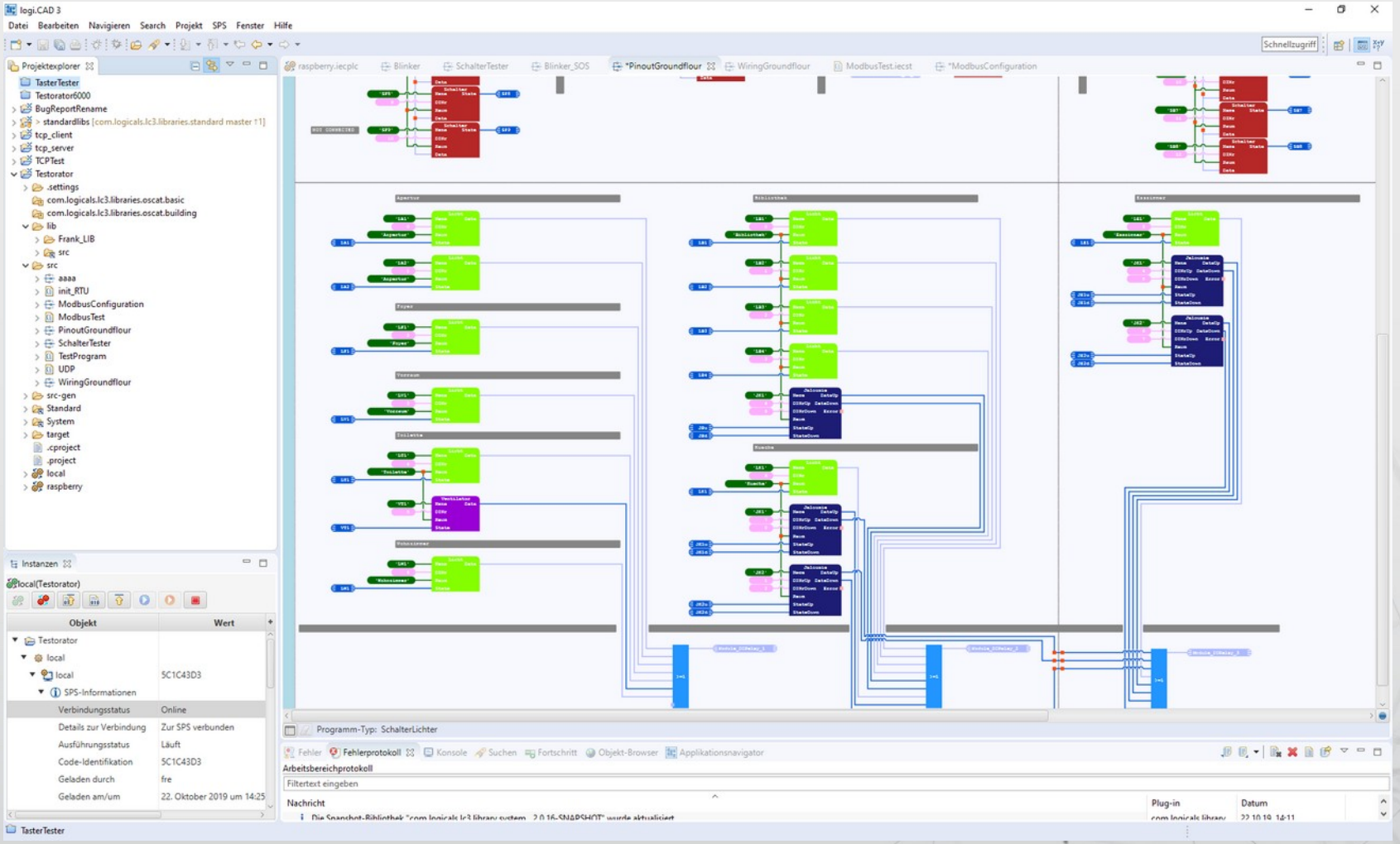
2
OFFICES

WE EMPOWER AUTOMATION

Overview: Our (Original) Goal

IDE for Programmable Logic Controllers

logi.cals



- **IEC 61131-3**
  - Structure Text (ST)
  - Function Block Diagram (FBD)
  - Sequential Function Chart (SFC)

Overview: Our (Original) Goal

IDE for Programmable Logic Controllers

logi.cals

WE EMPOWER AUTOMATION

Solutions
Overview

logi.cals

AUTOMOTIVE

80% of all buses in Europe

BUILDING

Thousands of controllers for security and comfort in modern ships

ENERGY

30% of all hydro-electric power plants

MOBILE AUTOMATION

Controllers for off-highway and agricultural machines

SYSTEM

Projects with 6 million I/O and communication data points

Continental

SIEMENS Building Technologies

ANDRITZ Hydro

STW

SMS group

WE EMPOWER AUTOMATION

logi.cals Live Demo: logi.CAD3

logi.cals Austria
Europaplatz 7/1
3100 St. Pölten
Austria
+43 5 77147-0
office@logicals.com

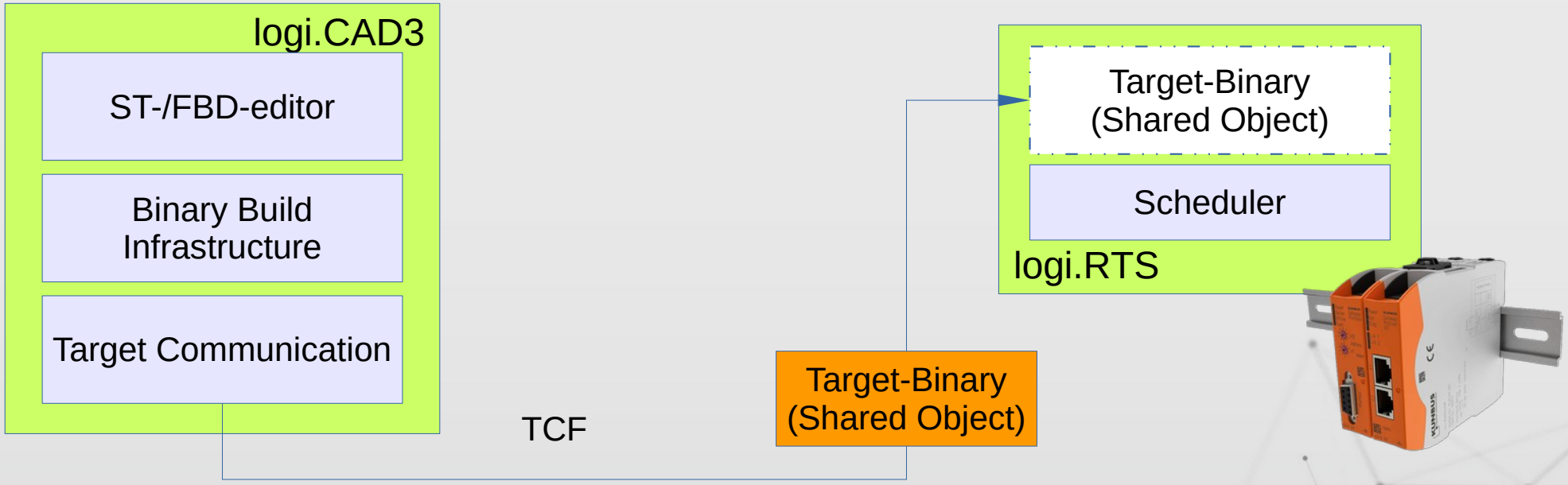
logi.cals Germany
Postfach 1306
40738 Langenfeld
Germany
+49 2173 9191-0
germany@logicals.com

www.logicals.com
Follow us on social media:

WE EMPOWER AUTOMATION

Overview: Current State
High-Level Architecture

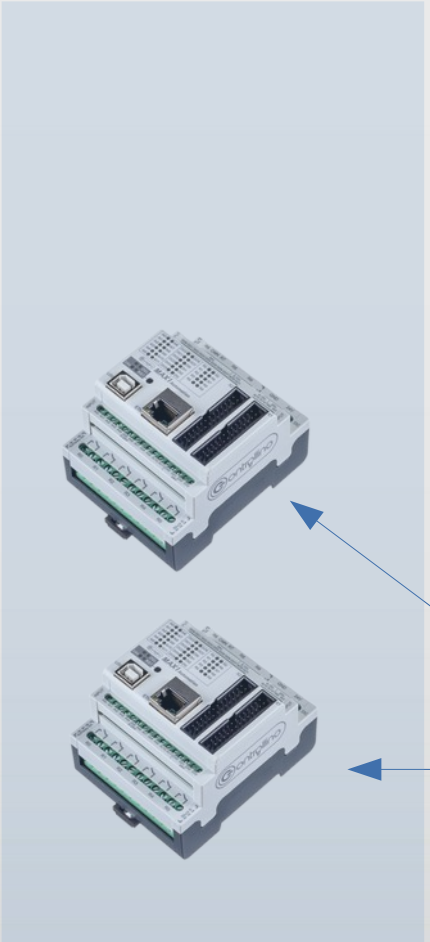
logi.cals



Architecture of logi.Cloud

Architecture and Components

logi.cals



logi.CAD
Build Server

logi.CAD
LC3 Server

logi.CAD
Target Comm Server

THEIA

Targets

Engineering

Theia IDE Backend

Theia IDE Frontend

logi.cals Live Demo: logi.CLOUD

logi.cals Austria
Europaplatz 7/1
3100 St. Pölten
Austria
+43 5 77147-0
office@logicals.com

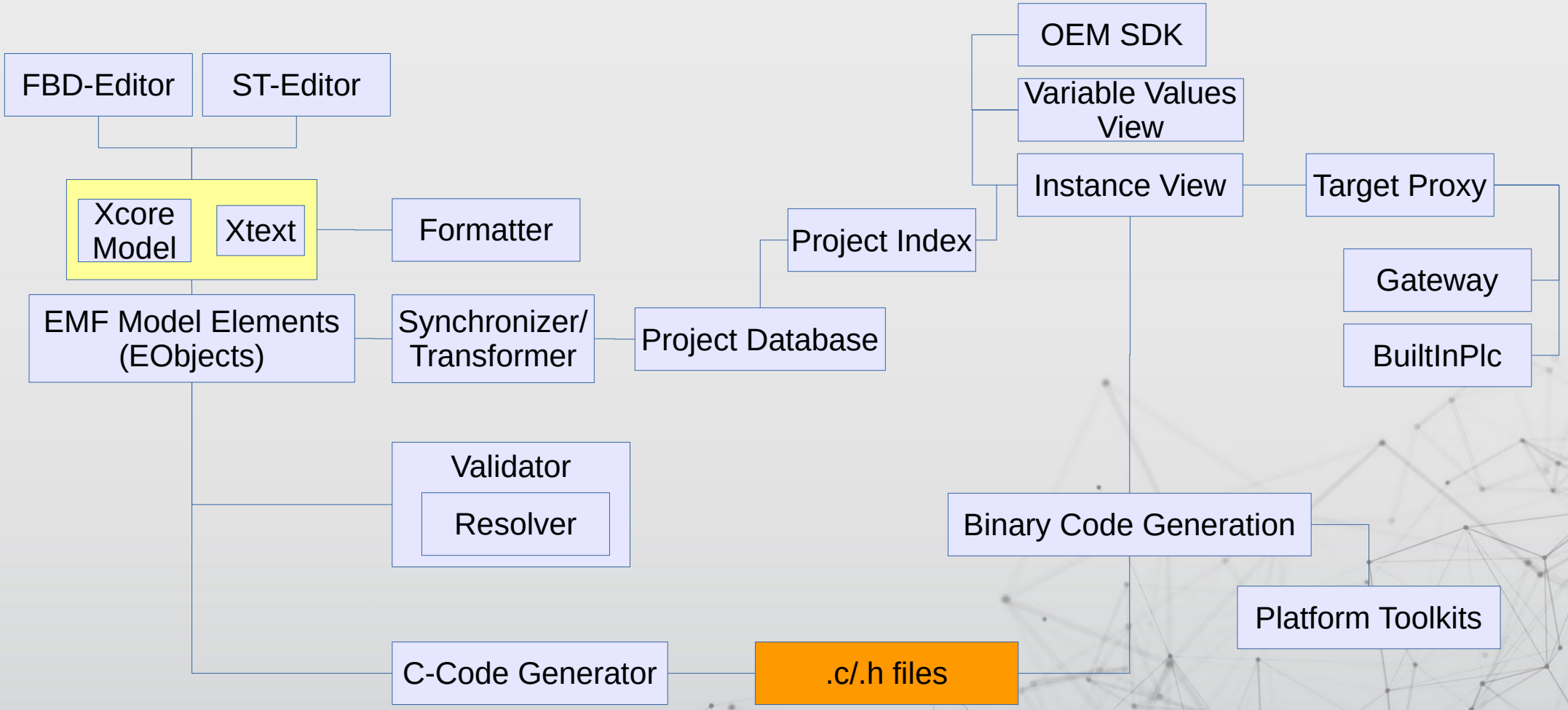
logi.cals Germany
Postfach 1306
40738 Langenfeld
Germany
+49 2173 9191-0
germany@logicals.com

www.logicals.com
Follow us on social media:

WE EMPOWER AUTOMATION

Structure of logi.CAD3
Architecture and Components

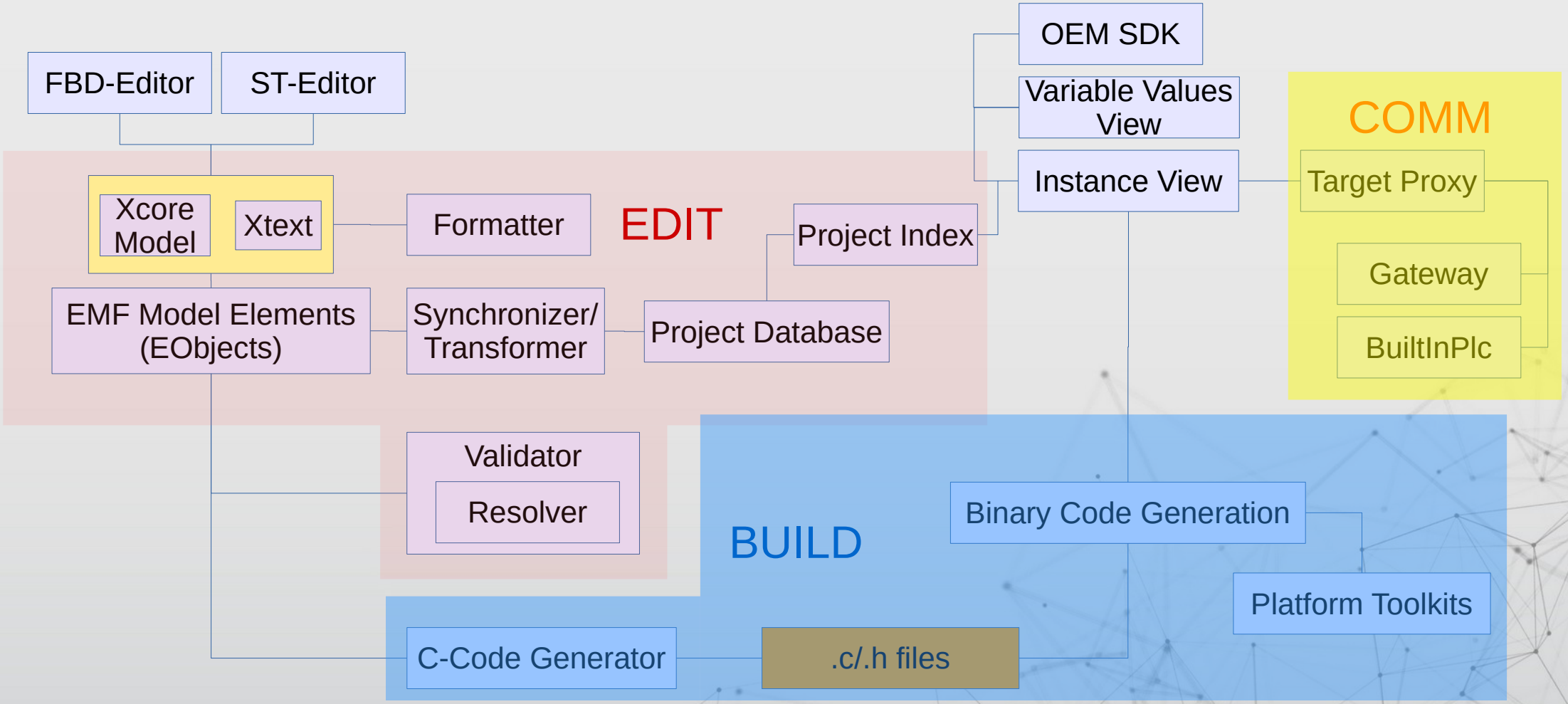
logi.cals



Structure of logi.CAD3

What will be re-used for the Cloud?

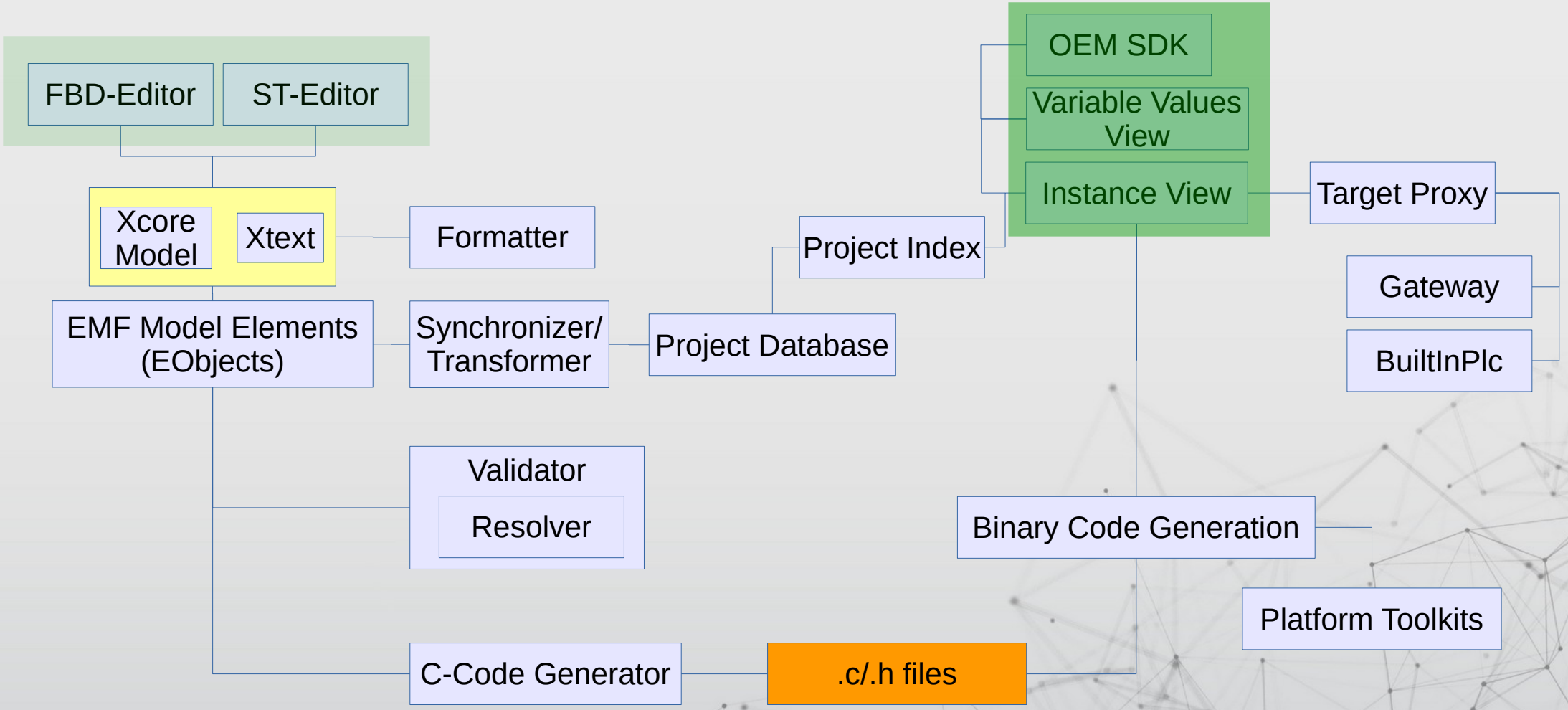
logi.cals



Structure of logi.CAD3

What will be re-used for the Cloud?

logi.cals



Architecture of logi.Cloud

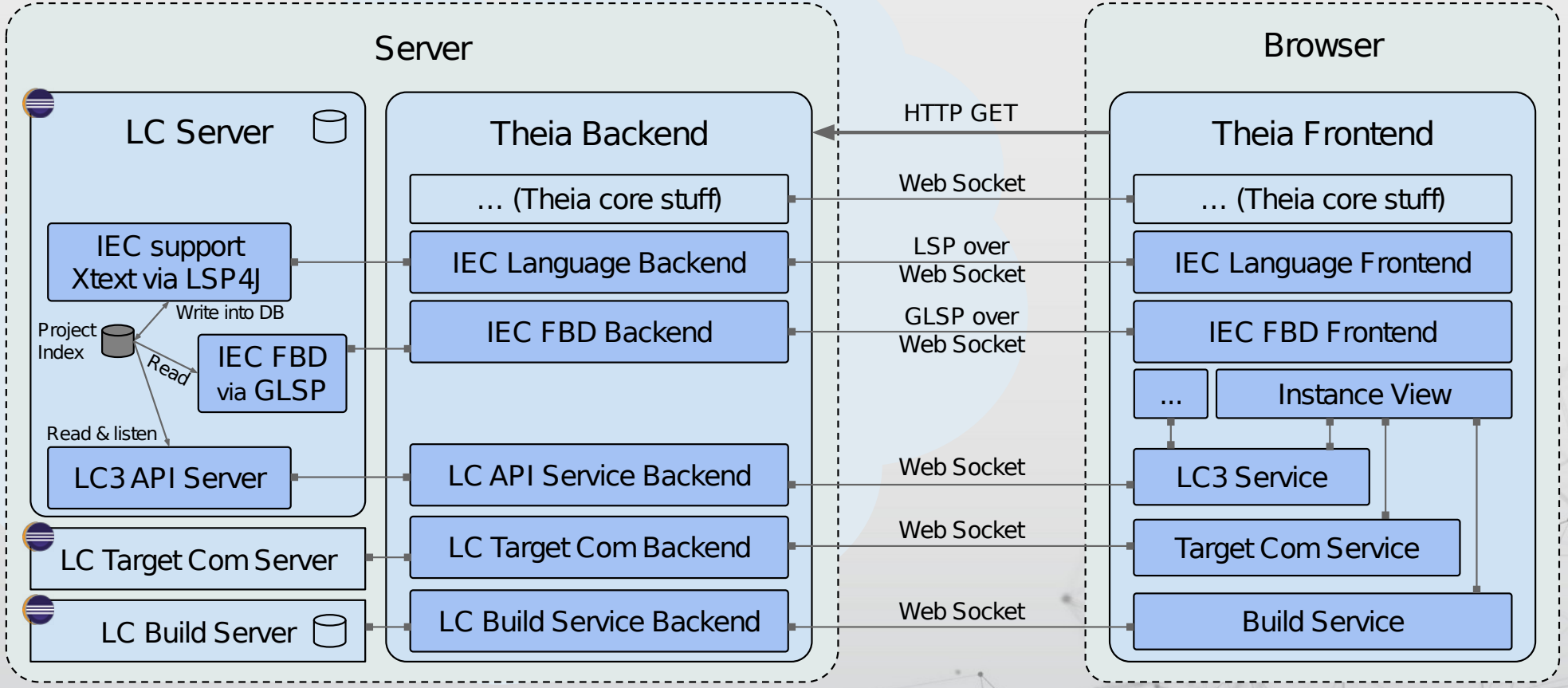
Architecture and Components

- **logi.CAD3 offers public API**
  - logi.CAD3 is a tool integrated by OEMs
  - API is pretty comprehensive
- **Makes it quite simple to**
  - Extend logi.CAD3 itself
  - Create stand-alone applications (with extensions)

Architecture of logi.Cloud

Architecture and Components

logi.cals



Key Enablers

Making logi.CAD3 ready for the Cloud

logi.cals

- **Eclipse projects**
  - Xtext: language mechanisms
  - LSP4J: JSON RPC over WebSockets (incl. support for LSP)
  - Eclipse Theia: Frontend/Backend framework
  - Eclipse GLSP: Graphical Language Server Protocol
  - Eclipse Che: future execution context of logi.CLOUD

Challenges: Dependency Categorization

Making logi.CAD3 ready for the Cloud

logi.cals

- **UI dependencies**
  - Rely on Eclipse UI to work
  - Example: custom editors
- **Eclipse dependencies**
  - Rely on common Eclipse functionality (but no UI)
  - Example: Eclipse builders, Workspace
- **OSGi**
  - Low-level Eclipse functionality
  - Example: Service access
- **OK**
  - Functionality that works in any Java environment
  - Example: (Generated) Xtext parsers

Challenges: Dependencies

Making logi.CAD3 ready for the Cloud

logi.cals

- **Dependency Analysis of Java Code using CDA**
  - **C**lass **D**ependency **A**nalyzer
    http://www.dependency-analyzer.org/
  - Handles class, package, and plug-in level dependencies
  - Can visualize dependency hierarchies
- **Dependencies exported to CSV format**
- **Categorization and aggregation in Spreadsheet Software**
  - Identification of (potentially) problematic dependencies

Challenges: Dependency Categorization

Making logi.CAD3 ready for the Cloud

logi.cals

			Direct dependency types (plugin)					Direct dependency types (class)			
14	Plugin	Included in headless	Eclipse	UI	OSGi	?		Eclipse	UI	OSGi	?
17	com.logicals.buildservice_2.6.0.201909060757.jar	Yes	2	1	2	0		12	2	7	0
18	com.logicals.common_2.6.0.201909060757.jar	Yes	2	2	4	5	wip	18	3	21	22
26	com.logicals.debug.common_2.6.0.201909060757.jar	Yes	4	4	4	2		16	8	14	26
37	com.logicals.iec.fbd_2.6.0.201909060757.jar	Yes	1	0	0	1		1	0	7	5
38	com.logicals.iec.model_2.6.0.201909060757.jar	Yes	2	1	3	1		9	4	10	1
39	com.logicals.iec.st_2.6.0.201909060757.jar	Yes	2	0	3	1		7	0	8	2
49	com.logicals.iec.st.ide_2.6.0.201909060757.jar	Yes	2	2	2	0		26	9	12	0
58	com.logicals.lc3.api_2.6.0.201909060757.jar	Yes	2	2	4	0	wip	5	3	17	0
52	com.logicals.lc3.api.model.st.impl_2.6.0.201909060757.jar	Yes	0	0	2	1	wip	0	0	2	1
53	com.logicals.lc3.api.mrc_2.6.0.201909060757.jar	Yes	1	0	2	1		4	0	4	1
54	com.logicals.lc3.help_2.6.0.201909060757.jar	Yes	1	2	2	3		1	7	5	9
70	com.logicals.library_2.6.0.201909060757.jar	Yes	1	0	3	1		9	0	13	4
77	com.logicals.mrc_2.6.0.201909060757.jar	Yes	2	4	4	1	wip	9	18	10	1
82	com.logicals.projectindex_2.6.0.201909060757.jar	Yes	2	1	3	1		11	2	18	7
84	com.logicals.targetproxy_2.6.0.201909060757.jar	Yes	1	2	3	1		1	3	8	10
87	com.logicals.uploader.lc3uploader_2.6.0.201909060757.jar	Yes	1	1	2	0		4	2	6	0

Challenges: Workspace handling

Making logi.CAD3 ready for the Cloud

logi.cals

- **Challenge:**
  - Eclipse Workspace depending functionality (Project Index; H2-based)
  - Theia IDE does not know Eclipse Workspaces
- **Solution:**
  - Custom “Cloud-aware” Workspace implementation
  - Will behave differently to Eclipse file system
  - Limits possibility of shared code with logi.CAD3

Challenges: Workspace handling

Making logi.CAD3 ready for the Cloud

logi.cals

- **Workspace types being used**
  - Theia Backend: FS/TS-based workspace
  - logi.CAD3 Server: Cloud-aware workspace
  - logi.CAD3 Build Server: Eclipse workspace (generated; snapshot)
  - logi.CAD3 Target Comm Server: none needed

Outlook
The (near) Future

logi.cals

- **Full support for C/C++ editing**
  - Based on clangd
- **Diagnostic Features**
  - View variable values
- **Continue FBD support**
  - Variable table
  - Context menus
  - Full editing, e. g. routing for connections
  - Copy-paste
  - ...

Outlook
The (far) Future

logi.cals

- **Integration of new platforms**
  - Licensing issues/challenges
- **Workspace management**
- Based on Eclipse Che
- **Decision on the execution platform for logi.CLOUD**
  - RedHat OpenShift, MS Azure, ...

Outlook
The (far) Future

logi.cals

- **Offering logi.CLOUD as an integrable solution**
  - logi.cals products typically sold as integrable framework
  - OEMs use our product and add branding
  - OEMs expect a plugin infrastructure

Summary
Findings, Recommendations

logi.cals

- **Build Proof-of-Concept Prototypes**
  - Based on well-defined use-cases
  - Getting to know new technologies and frameworks
  - Helps to estimate the time frame for developing the actual product

Bibliography
Further reading

logi.cals

- **Overall Architecture**
  - (Initial) Eclipse Theia Design Document:

    https://docs.google.com/document/d/1aodR1LJEF_zu7xBis2MjpHRyv7JKJzW7EWI9XRYCt48/
  - EclipseSource Documents on Eclipse Theia and Eclipse Che:

    https://eclipsesource.com/blogs
- **Getting Started (Motivational example)**
  - https://theia-ide.org/docs/

logi.cals Thank you for your attention!

logi.cals Austria
Europaplatz 7/1
3100 St. Pölten
Austria
+43 5 77147-0
office@logicals.com

logi.cals Germany
Postfach 1306
40738 Langenfeld
Germany
+49 2173 9191-0
germany@logicals.com

www.logicals.com
Follow us on social media:

WE EMPOWER AUTOMATION

logi.cals Contact

logi.cals Austria
Europaplatz 7/1
3100 St. Pölten
Austria
+43 5 77147-0
office@logicals.com

logi.cals Germany
Postfach 1306
40738 Langenfeld
Germany
+49 2173 9191-0
germany@logicals.com

www.logicals.com
Follow us on social media:

WE EMPOWER AUTOMATION

logi.cals Spare Slide(s)

logi.cals Austria
Europaplatz 7/1
3100 St. Pölten
Austria
+43 5 77147-0
office@logicals.com

logi.cals Germany
Postfach 1306
40738 Langenfeld
Germany
+49 2173 9191-0
germany@logicals.com

www.logicals.com
Follow us on social media:

WE EMPOWER AUTOMATION

Challenges: Additional concerns

Making logi.CAD3 ready for the Cloud

logi.cals

LC3/logi.CLOUD: General interface of LC3 projects:

```
public interface Lc3Project extends Comparable<Lc3Project> {
  [...]
  File getFile(String name);
}
```

Challenges: Additional concerns

Making logi.CAD3 ready for the Cloud

logi.cals

LC3: Using Eclipse `IProject` to retrieve files

```
@Override
public File getFile(final String name) {
  final IFile eclipseFile = this.eclipseProject.getFile(name);
  return new File(eclipseFile.getLocation().toOSString());
}
```

logi.CLOUD: Using Language Server Project Manager to retrieve files (LC3 Server; JAVA)

```
@Override
public File getFile(final String name) {
  final URI baseDir = this.projectManager.getBaseDir();
  return new File(baseDir.toFileString() + name);
}
```

Challenges: Additional concerns

Making logi.CAD3 ready for the Cloud

logi.cals

LC3: Using Eclipse Workspace and Projects to initialize Project Index

```
private void initializeProjects() {
  final IProject[] iProjects = ResourcesPlugin.getWorkspace().getRoot().getProjects();
  for (final IProject iProject : iProjects) {
    if (iProject.isOpen() && IECProjectNature.isIECProject(iProject)) {
      final Lc3Project project = new Lc3ProjectImpl(iProject);
      projects.put(project.getName(), project);
    }
  }
}
```

Challenges: Additional concerns

Making logi.CAD3 ready for the Cloud

logi.cals

logi.CLOUD: Use Language Server Workspace Manager and Project Manager

to initialize Project Index (LC3 Server; JAVA)

```
private void initializeProjects() {
  for (final ProjectManager projectManager : workspaceManager.getProjectManagers()) {
    if (projectManager.getBaseDir() != null) {
      final Lc3Project project = new LC3CloudProject(projectManager);
      projects.put(project.getName(), project);
    }
  }
}
```

Challenges: Additional concerns

Making logi.CAD3 ready for the Cloud

logi.cals

LC3: Build and Upload Program triggered by client

```
protected void buildAndLoadApplication(final ActionEvent event) {
  TargetSession targetSession = TargetProxy.getTargetSessionForUI(resource);
  new BuildProgramJob(targetSession.getResource(), new BuildOptions()).schedule();
  new UploadProgramJob(targetSession).schedule();
}

public static NormalizedFile getTargetOutputDirectory(IProject project) {
  return new NormalizedFile(project.getLocation().toFile()).append("target");
}
```

Challenges: Additional concerns

Making logi.CAD3 ready for the Cloud

logi.cals

logi.CLOUD: Build and Upload Program triggered by client (Theia Backend; TypeScript)

```
async buildAndLoadApplication(connectionInfo: Lc3ResourceConnectionInfo): Promise<void> {
  const jobId = await this.buildService.buildWithProgress(connectionInfo.resource.instancePath);
  await this.buildService.onJobFinished(jobId);
  const buildLocation = await this.buildService.getBuildLocation();
  return this.targetService.upload(connectionInfo, buildLocation);
}
```

logi.CLOUD: Determining the actual build location (LC3 Build Server; JAVA)

```
public NormalizedFile getTargetOutputDirectory(String buildLocation, String projectName) {
  return new NormalizedFile(buildLocation).append(projectName).append("target");
}
```

Overview: PLC-Programming

Programming Languages: ST (IEC61131-3)

logi.cals