Building Graphical Web Applications: A Case Study on SysON

24 Oct 2024

Axel RICHARD
Web & Modeling Consultant
axel.richard@obeo.fr | @_axelrichard_

OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

Objectives

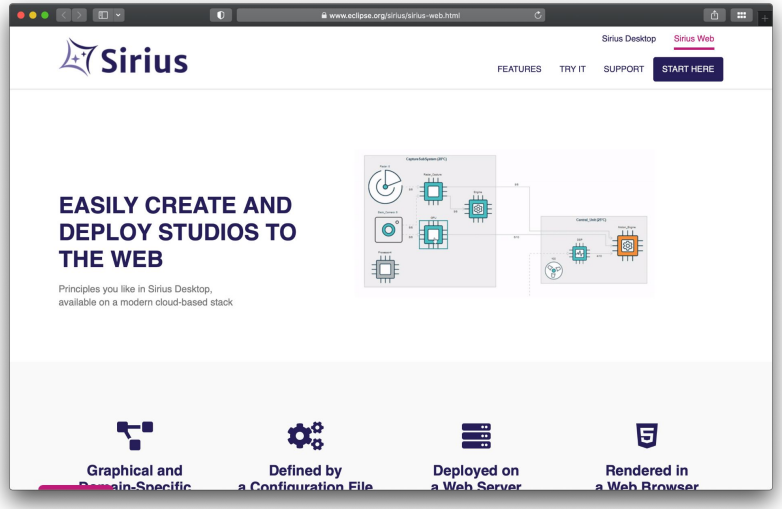
- Show you what is SysON
- A concrete example of a Sirius Web based application
- How to utilize Sirius Web's extension/customization capabilities

Contents

- Sirius Web
- SysON
- SysML v2
- SysON Demo
- Sirius Web API Builder
- Sirius Web Extensibility
- Sirius Web Customization
- Tests in SysON

Sirius Web

- Everything you liked in Sirius Desktop, available on a modern cloud-based stack
  - Eclipse Public License (EPL), open-source
  - Graphical and Domain specific tooling
  - Defined by a configuration file
  - Deployed on a web server
  - Rendered in a web browser
  - Collaborative support



What’s inside Sirius Web?

READY-TO-USE

Modeling framework to define and render graphical applications in the web

What's inside Sirius Web?

READY-TO-USE

Modeling framework to define and render graphical applications in the web

MODEL SERVER

Open source model server components with a GraphQL API

What's inside Sirius Web?

READY-TO-USE

Modeling framework to define and render graphical applications in the web

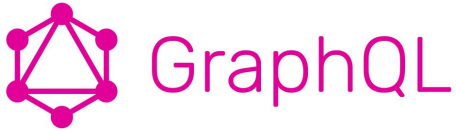
MODEL SERVER

Open source model server components with a GraphQL API

MODEL APPLICATION

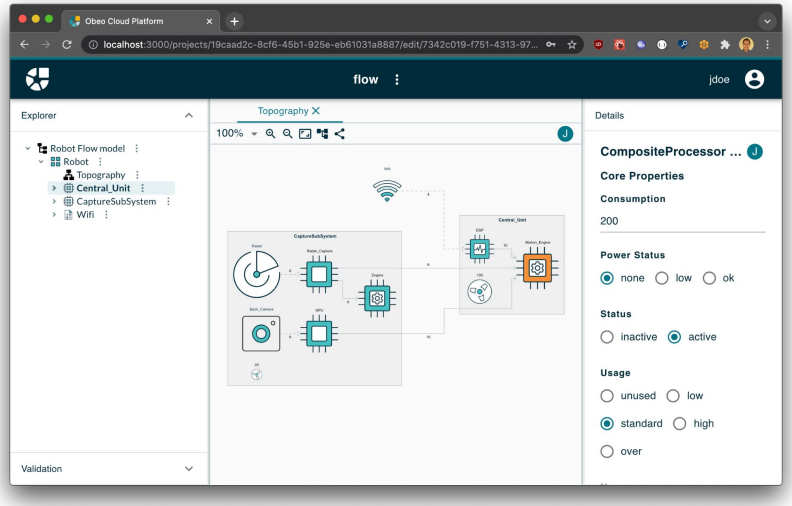
Open source model application (diagram, properties, forms...)

Built on top of awesome technologies



Obeo Cloud Platform

- All of Sirius Web with additional collaborative and access control features
  - Authentication and authorization
  - Public/Private projects
  - Role based access control
  - Indicators of active users
  - History of versions



OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

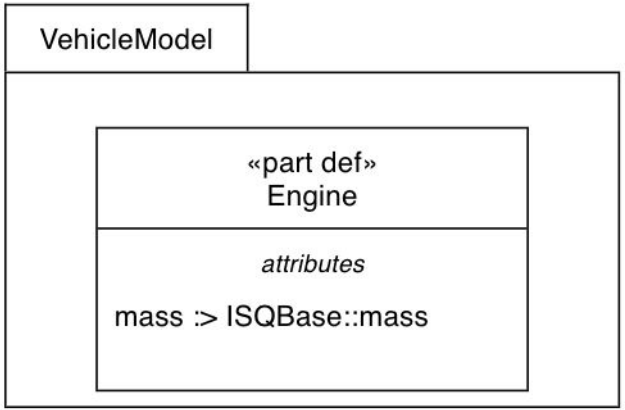
What is SysON

- Web-based tool built on top of Sirius Web
- Open-source & free, EPL
- Graphical modeling environment for SysML v2
- Features for creating, editing, and visualizing SysML v2 models
- Co-lead by Obeo and CEA

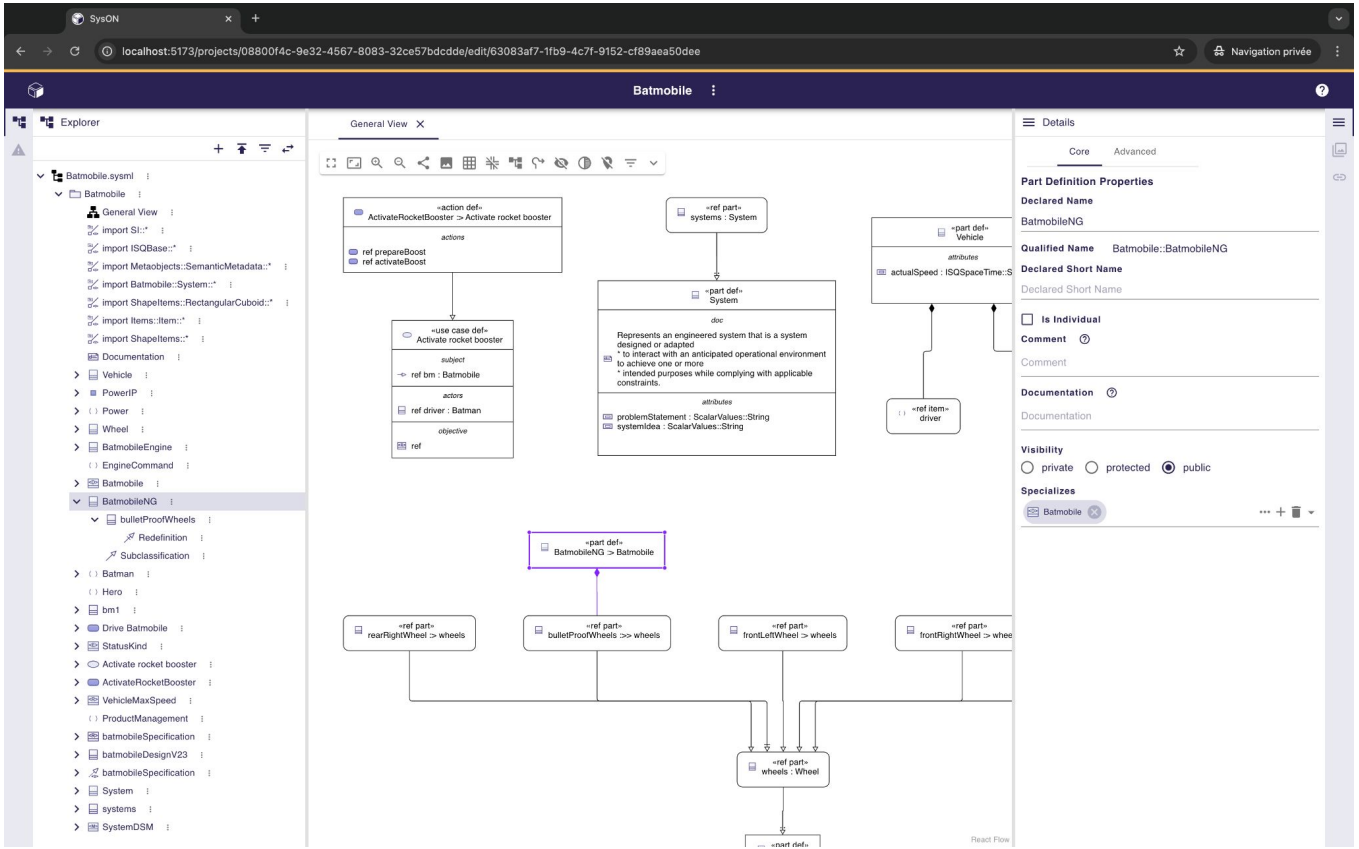
What is SysML v2

- A standardized language for modeling complex systems
  - defines a textual notation
  - defines a graphical notation
- Standardized at the OMG
  - https://www.omg.org/spec/SysML

```
package VehicleModel {
    part def Engine {
        attribute mass :> ISQBase::mass;
    }
}
```



SysON Demo

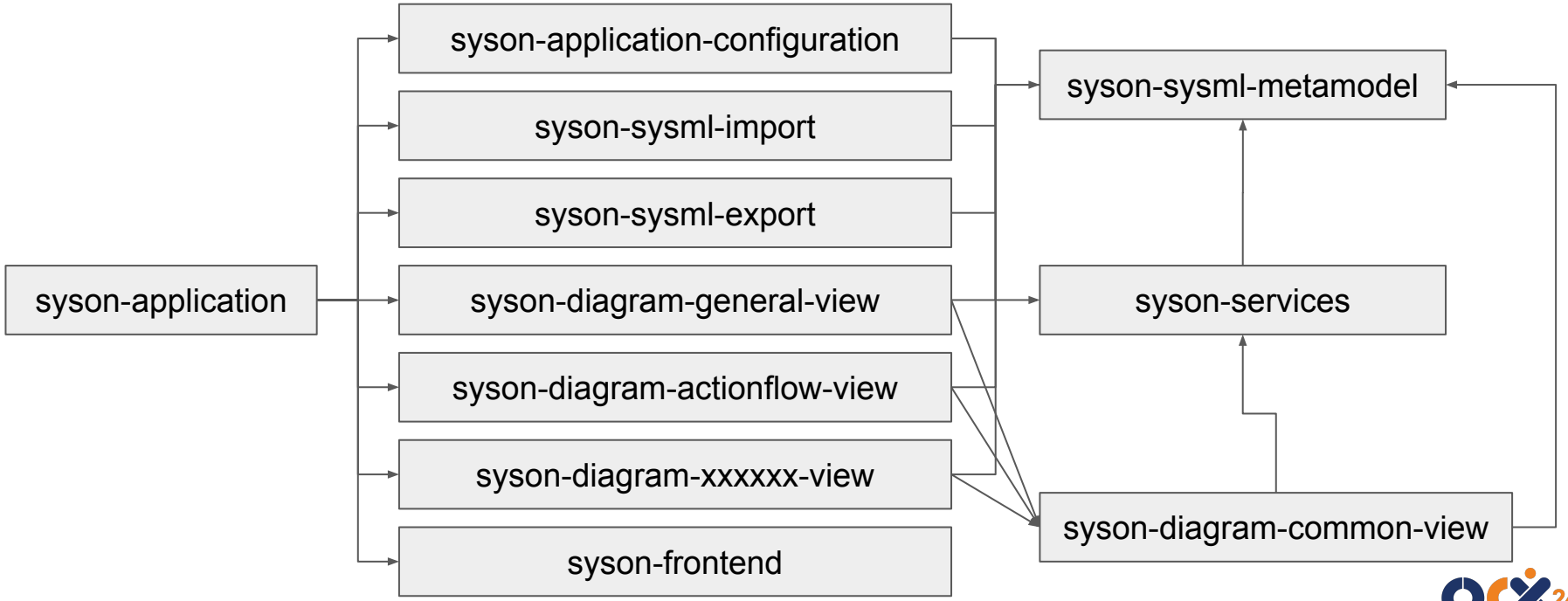


OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

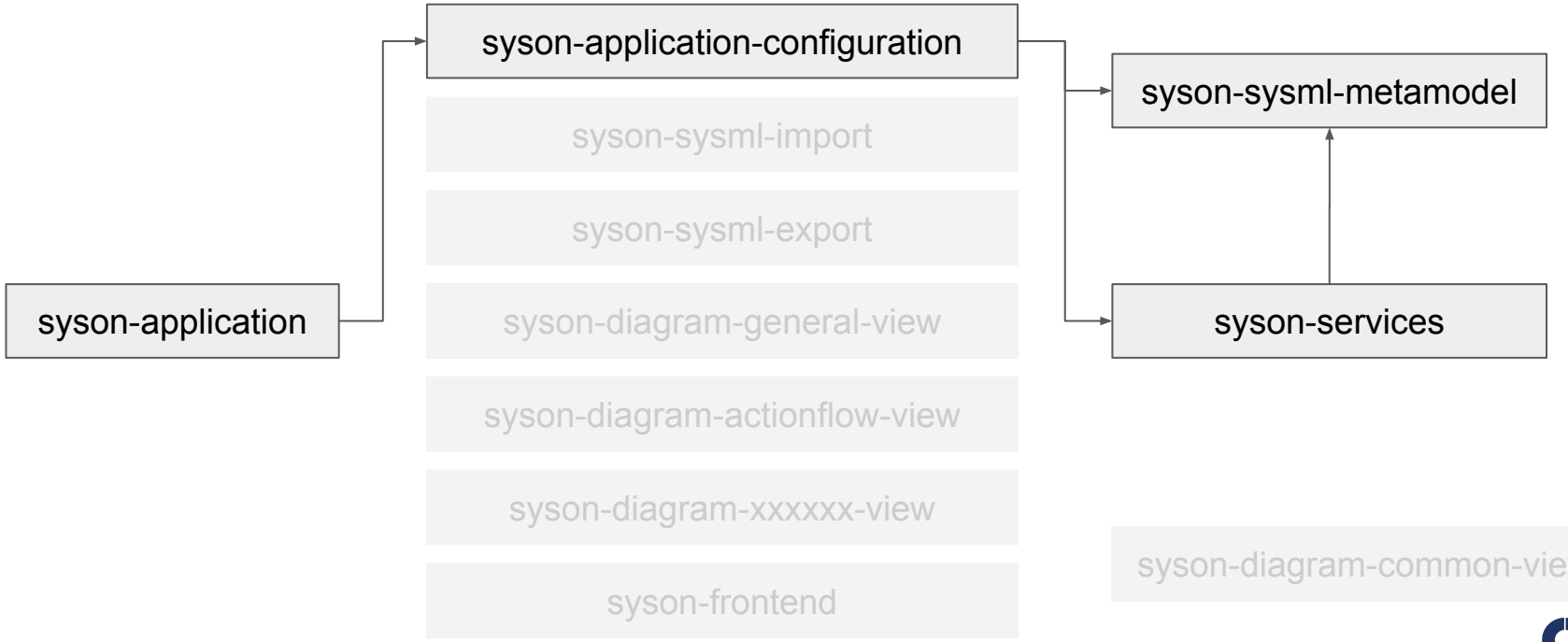
SysON codebase structure

- Backend



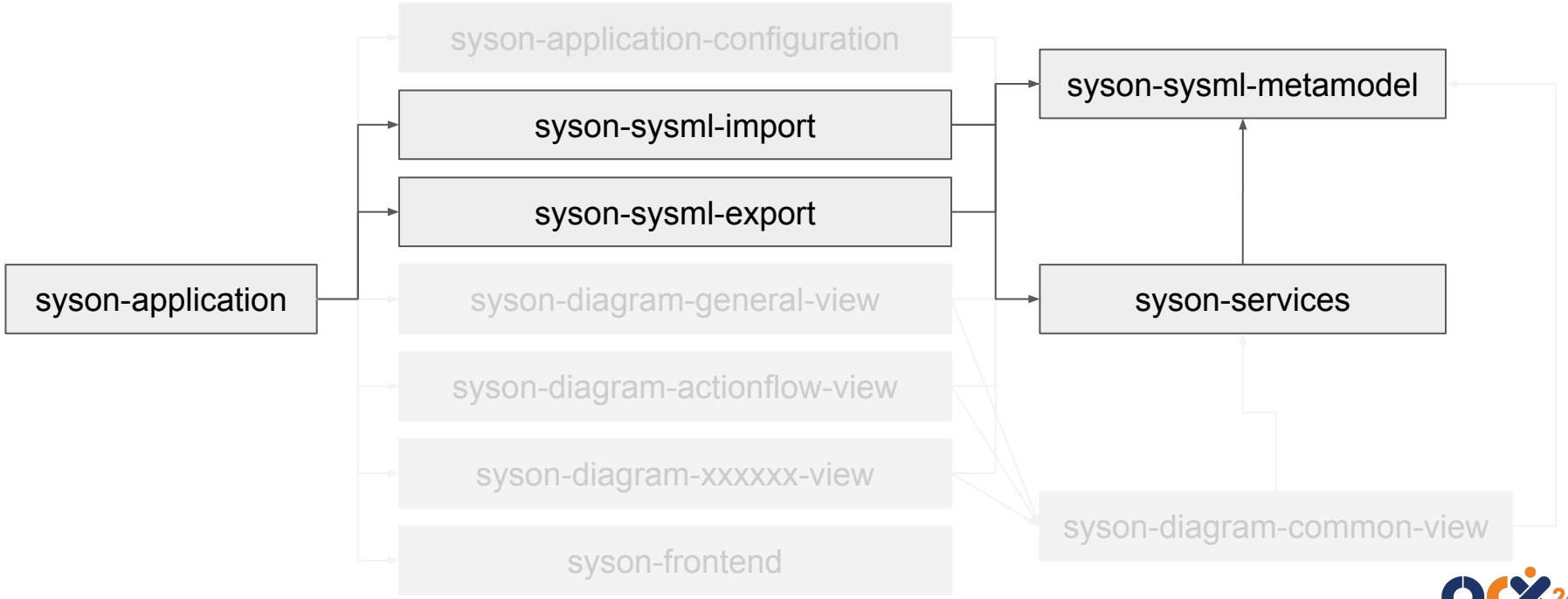
SysON codebase structure

- Backend



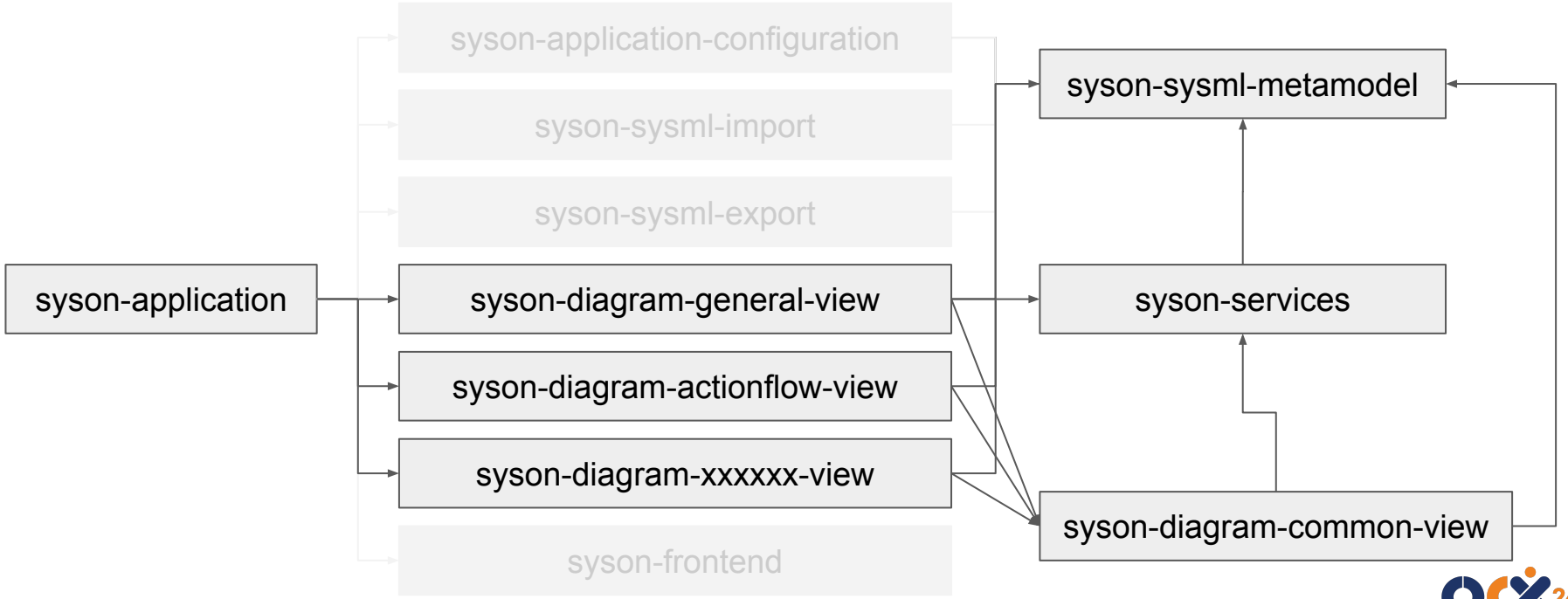
SysON codebase structure

- Backend



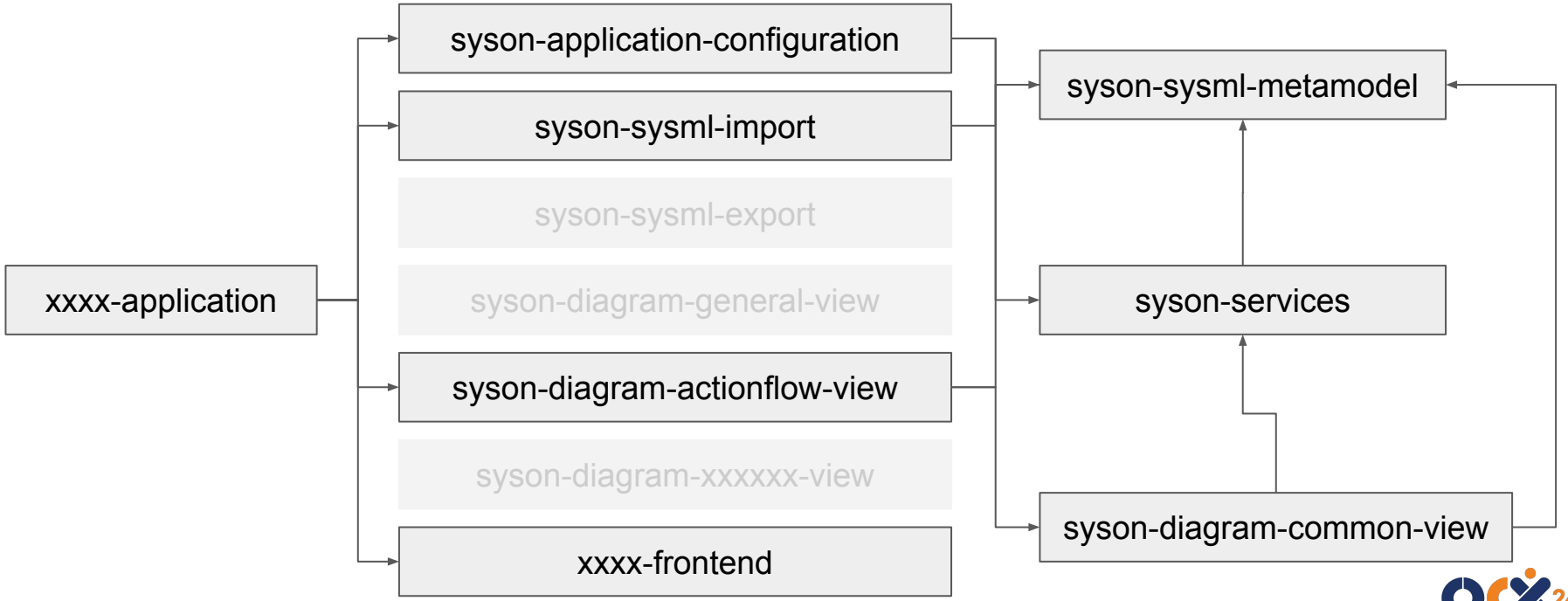
SysON codebase structure

- Backend



SysON codebase structure

- Backend

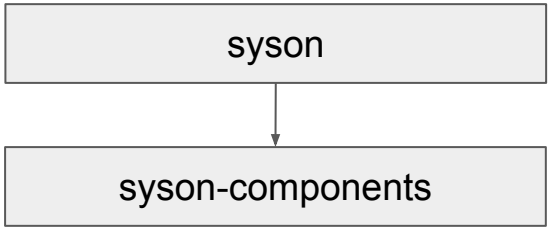


OBEO

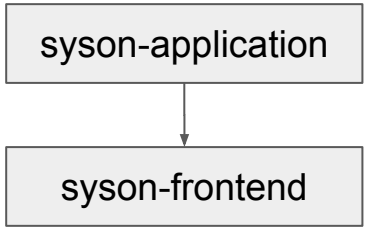
OCX24 OPEN COMMUNITY EXPERIENCE

SysON codebase structure

- Frontend



- Backend



Sirius Web API Builder

- Sirius Web diagrams are conform to diagram descriptions
- A diagram description is a kind of representation description
- API Builder facilitates the writing of representation descriptions

API Builder in SysON

- General View Diagram Description

GeneralViewDiagramDescriptionProvider.java

```java
import org.eclipse.sirius.components.view.builder.generated.diagram.DiagramBuilders;

private final DiagramBuilders diagramBuilderHelper = new DiagramBuilders();

@Override
public RepresentationDescription create(IColorProvider colorProvider) {
  var domainType = SysMLMetamodelHelper.buildQualifiedName(SysmlPackage.eINSTANCE.getNamespace());
  var diagramDescription = this.diagramBuilderHelper.newDiagramDescription()
            .arrangeLayoutDirection(ArrangeLayoutDirection.DOWN)
            .autoLayout(false)
            .domainType(domainType)
            .preconditionExpression(AQLUtils.getSelfServiceCallExpression("canCreateDiagram"))
            .name(DESCRIPTION_NAME)
            .titleExpression(DESCRIPTION_NAME)
            .build();
  ...
  return diagramDescription;
}
```

CodeImage

OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

API Builder in SysON

- Package Node Description

PackageNodeDescriptionProvider.java

```java
import org.eclipse.sirius.components.view.builder.generated.diagram.DiagramBuilders;

private final DiagramBuilders diagramBuilderHelper = new DiagramBuilders();

@Override
public NodeDescription create() {
    var domainType = SysMLMetamodelHelper.buildQualifiedName(SysmlPackage.eINSTANCE.getPackage());
    return this.diagramBuilderHelper.newNodeDescription()
            .collapsible(true)
            .childrenLayoutStrategy(new FreeFormLayoutStrategyDescriptionBuilder().build())
            .defaultHeightExpression("100")
            .defaultWidthExpression("300")
            .domainType(domainType)
            .insideLabel(this.createInsideLabelDescription())
            .name(this.nameGenerator.getNodeName(SysmlPackage.eINSTANCE.getPackage()))
            .semanticCandidatesExpression(AQLUtils.getSelfServiceCallExpression("getAllReachable", domainType))
            .style(this.createPackageNodeStyle())
            .userResizable(UserResizableDirection.BOTH)
            .synchronizationPolicy(SynchronizationPolicy.UNSYNCHRONIZED)
            .build();
}
```

CodeImage

API Builder in Sirius Web

- Form Description

GeneralViewDiagramDescriptionProvider.java

```java
import org.eclipse.sirius.components.view.builder.generated.from.FormBuilders;
import org.eclipse.sirius.components.view.builder.generated.view.ViewBuilders;

private FormDescription createFormDescription() {
    var defaultStyle = new FormBuilders().newRadioDescriptionStyle()
            .build();

    var editRadio = new ChangeContextBuilder()
            .expression("aql:self")
            .children(new ViewBuilders().newUnsetValue()
                            .featureName("superTypes")
                            .elementExpression("aql:self.superTypes")
                            .build(),
                    new ViewBuilders().newSetValue()
                            .featureName("superTypes")
                            .valueExpression("aql:newValue")
                            .build())
            .build();

    var radioDescription = new FormBuilders().newRadioDescription()
            .name("Checkbox")
            .labelExpression("aql:'SuperType'")
            .helpExpression("Pick a super type")
            .valueExpression("aql:self.superTypes->first()")
            .candidatesExpression("aql:self.eContainer(domain::Domain).types")
            .candidateLabelExpression("aql:candidate.name")
            .style(defaultStyle)
            .body(editRadio)
            .build();
    ...
```

CodeImage

GeneralViewDiagramDescriptionProvider.java

```java
    ...
    var groupDescription = new GroupDescriptionBuilder()
            .name("Group")
            .labelExpression("Group")
            .semanticCandidatesExpression("aql:self")
            .children(radioDescription)
            .build();

    var pageDescription = new PageDescriptionBuilder()
            .name("Page")
            .labelExpression("Page")
            .domainType("domain:Entity")
            .semanticCandidatesExpression("aql:self")
            .groups(groupDescription)
            .build();

    this.formDescription = new FormDescriptionBuilder()
            .name("Form")
            .titleExpression("aql:'FormWithRadio'")
            .domainType("domain:Entity")
            .pages(pageDescription)
            .build();
    return this.formDescription;
}
```

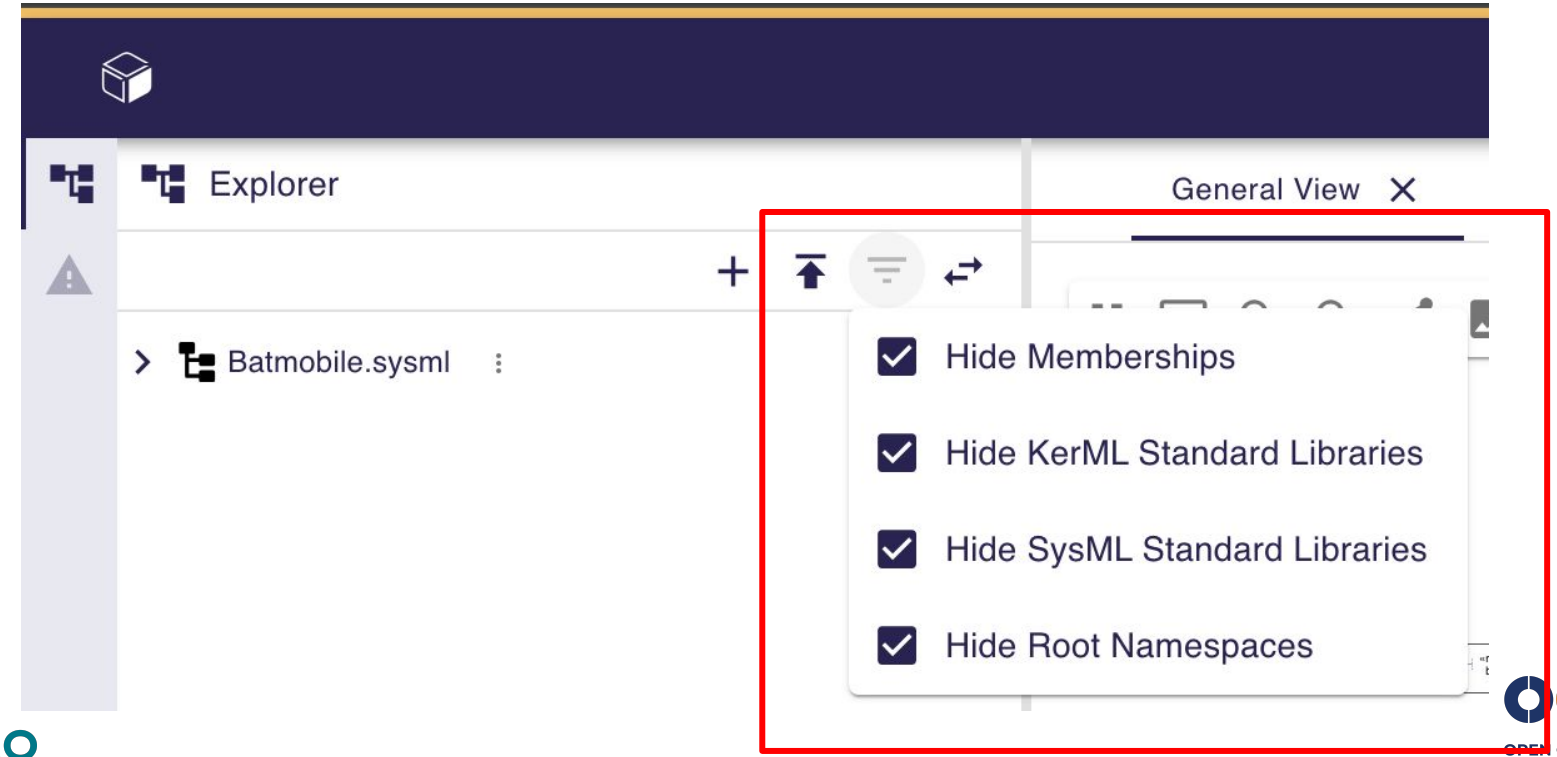
CodeImage

OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

Sirius Web Extensibility

- Explorer Filters



Sirius Web Extensibility

- Explorer Filters

HideRootNamespaceTreeItemAlteredContentProvid...

```java
import org.eclipse.sirius.web.application.views.explorer.services.api.IExplorerTreeItemAlteredContentProvider;

@Service
public class HideRootNamespaceTreeItemAlteredContentProvider implements IExplorerTreeItemAlteredContentProvider {

    private final UtilService utilService = new UtilService();

    @Override
    public boolean canHandle(Object object, List<String> activeFilterIds) {
        return activeFilterIds.contains(SysONTreeFilterProvider.HIDE_ROOT_NAMESPACES_ID);
    }

    @Override
    public List<Object> apply(List<Object> computedChildren, VariableManager variableManager) {
        var alteredChildren = new ArrayList<>();
        computedChildren.forEach(child -> {
            if (child instanceof Namespace namespace && this.utilService.isRootNamespace(namespace)) {
                alteredChildren.addAll(namespace.getOwnedElement());
            } else {
                alteredChildren.add(child);
            }
        });
        return alteredChildren;
    }
}
```

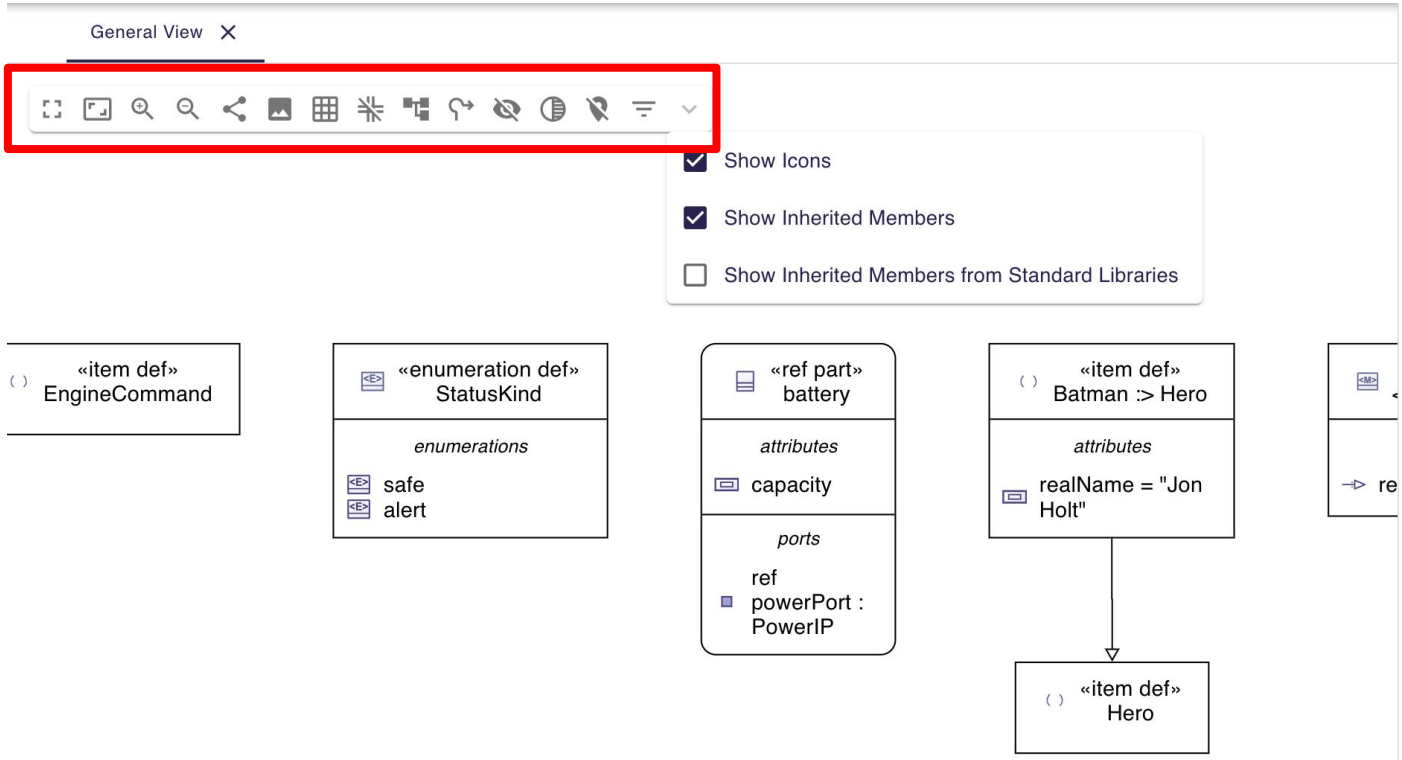
CodeImage

OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

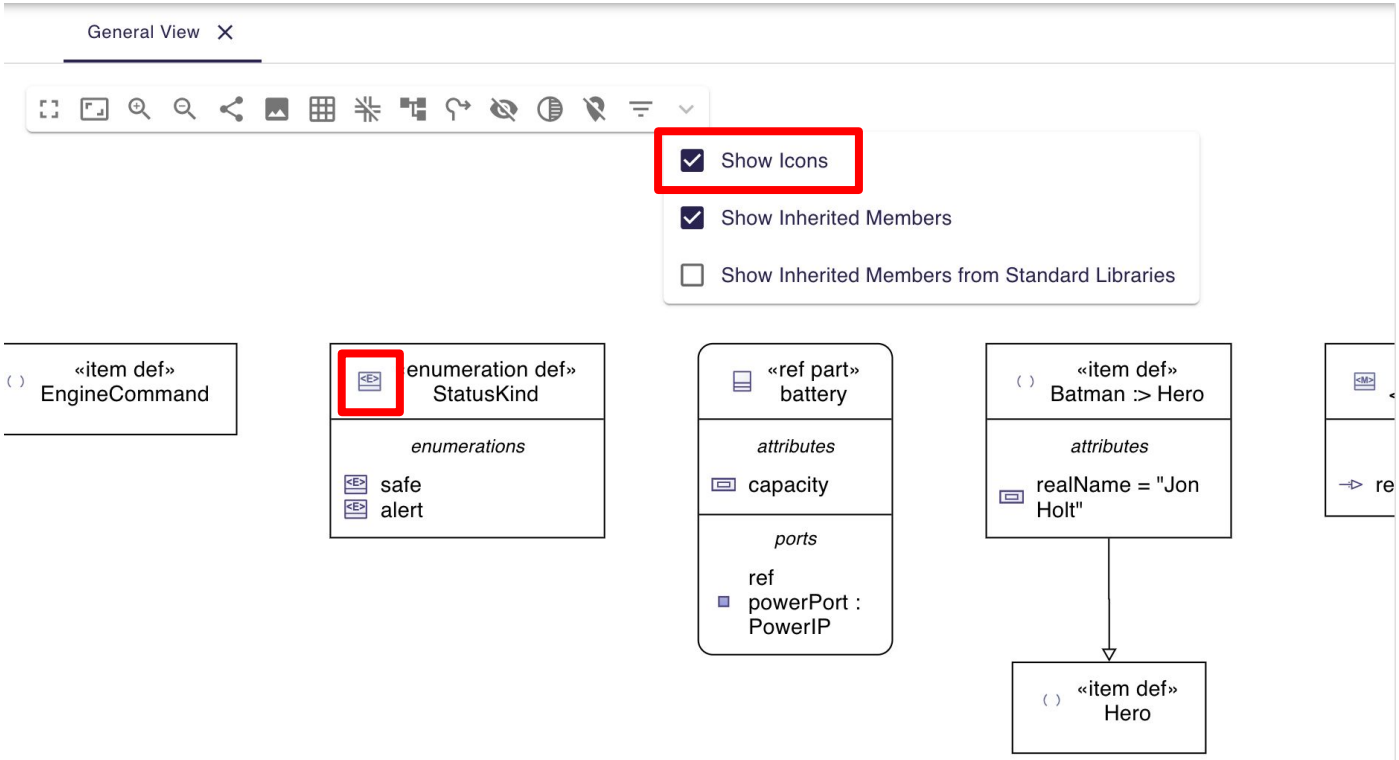
Sirius Web Extensibility

- Diagram Panel



Sirius Web Extensibility

- Diagram Panel

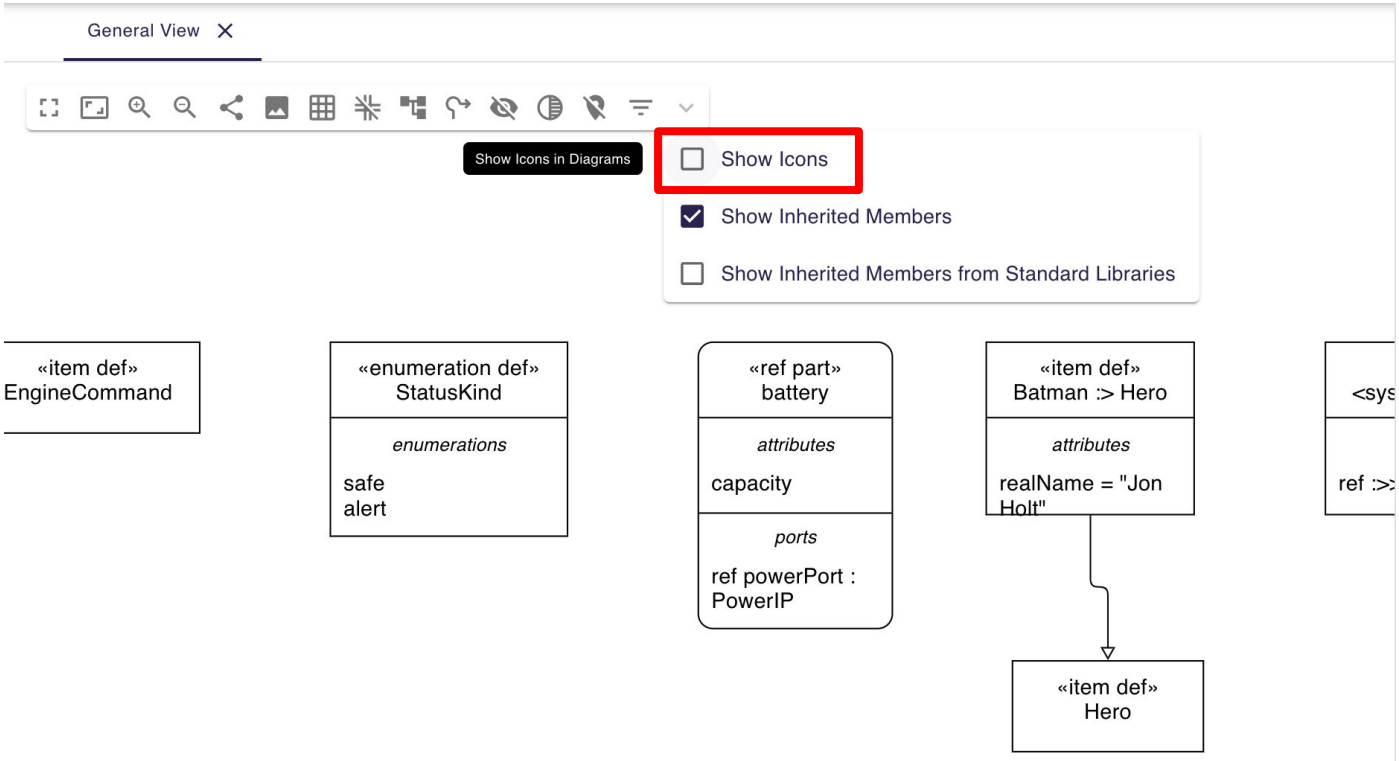


OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

Sirius Web Extensibility

- Diagram Panel



OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

Sirius Web Extensibility

- Diagram Panel - Frontend

ShowHideDiagramsIcons.tsx

```tsx
import { gql, useMutation, useQuery } from '@apollo/client';

const showDiagramsIconsQuery = gql`
  query getShowDiagramsIcons {
    viewer {
      showDiagramsIconsValue
    }
  }
`;

const setShowDiagramsIconsMutation = gql`
  mutation showDiagramsIcons($input: ShowDiagramsIconsInput!) {
    showDiagramsIcons(input: $input) {
      __typename
      ... on ShowDiagramsIconsSuccessPayload {
        show
      }
      ... on ErrorPayload {
        message
      }
    }
  }
`;

export const ShowHideDiagramsIcons = ({ editingContextId, diagramId }: DiagramPanelActionProps) => {
  ...
  return (
    <Tooltip title={state.tooltip} placement="left">
      <Checkbox checked={state.checked === null ? state.checked : true} onChange={handleChange} />
    </Tooltip>
  );
};
```

CodeImage

OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

Sirius Web Extensibility

- Diagram Panel - Backend

ShowDiagramsIconsDataFetcher.java

```java
import org.eclipse.sirius.components.annotations.spring.graphql.QueryDataFetcher;
import org.eclipse.sirius.components.graphql.api.IDataFetcherWithFieldCoordinates;
import org.eclipse.syson.diagram.common.view.services.ShowDiagramsIconsService;

import graphql.schema.DataFetchingEnvironment;

@QueryDataFetcher(type = "Viewer", field = "showDiagramsIconsValue")
public class ShowDiagramsIconsDataFetcher implements IDataFetcherWithFieldCoordinates<Boolean> {

    private final ShowDiagramsIconsService showDiagramsIconsService;

    public ShowDiagramsIconsDataFetcher(ShowDiagramsIconsService showDiagramsIconsService) {
        this.showDiagramsIconsService = Objects.requireNonNull(showDiagramsIconsService);
    }

    @Override
    public Boolean get(DataFetchingEnvironment environment) throws Exception {
        return Boolean.valueOf(this.showDiagramsIconsService.getShowIcons());
    }
}
```

CodeImage

Sirius Web Extensibility

- Diagram Panel - Backend

MutationShowDiagramsIconsDataFetcher.java

```java
import com.fasterxml.jackson.databind.ObjectMapper;

import org.eclipse.sirius.components.annotations.spring.graphql.QueryDataFetcher;
import org.eclipse.sirius.components.core.api.IPayload;
import org.eclipse.sirius.components.graphql.api.IEditingContextDispatcher;
import org.eclipse.sirius.components.graphql.api.IExceptionWrapper;

@QueryDataFetcher(type = "Mutation", field = "showDiagramsIcons")
public class MutationShowDiagramsIconsDataFetcher implements IDataFetcherWithFieldCoordinates<CompletableFuture<IPayload>> {

    private static final String INPUT_ARGUMENT = "input";

    private final ObjectMapper objectMapper;

    private final IExceptionWrapper exceptionWrapper;

    private final IEditingContextDispatcher editingContextDispatcher;
...
    @Override
    public CompletableFuture<IPayload> get(DataFetchingEnvironment environment) throws Exception {
        Object argument = environment.getArgument(INPUT_ARGUMENT);
        var input = this.objectMapper.convertValue(argument, ShowDiagramsIconsInput.class);

        return this.exceptionWrapper.wrapMono(() -> this.editingContextDispatcher.dispatchMutation(input.editingContextId(), input), input).toFuture();
    }
}
```

CodeImage

Sirius Web Extensibility

- Diagram Panel - Backend

ShowDiagramsIconsEventHandler.java

```java
import org.eclipse.sirius.components.collaborative.diagrams.api.IDiagramEventHandler;
import org.eclipse.sirius.components.core.api.IPayload;
import org.eclipse.syson.diagram.common.view.services.ShowDiagramsIconsService;

@Service
public class ShowDiagramsIconsEventHandler implements IDiagramEventHandler {

    private final ShowDiagramsIconsService showDiagramsIconsService;
    ...
    @Override
    public boolean canHandle(IDiagramInput diagramInput) {
        return diagramInput instanceof ShowDiagramsIconsInput;
    }

    @Override
    public void handle(One<IPayload> payloadSink, Many<ChangeDescription> changeDescriptionSink, IEditingContext editingContext, IDiagramContext diagramContext, IDiagramInput diagramInput) {
        ...
        this.showDiagramsIconsService.setShowIcons(showDiagramsIconsInput.show());
        IPayload payload = new ShowDiagramsIconsSuccessPayload(diagramInput.id(), showDiagramsIconsInput.show());
        ChangeDescription changeDescription = new ChangeDescription(DiagramChangeKind.DIAGRAM_ELEMENT_VISIBILITY_CHANGE, diagramInput.representationId(), diagramInput);
        payloadSink.tryEmitValue(payload);
        changeDescriptionSink.tryEmitNext(changeDescription);
        ...
    }
}
```

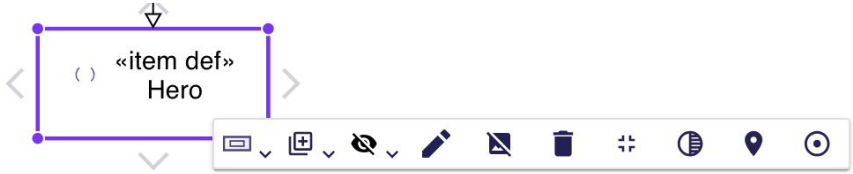
CodeImage

OBEO

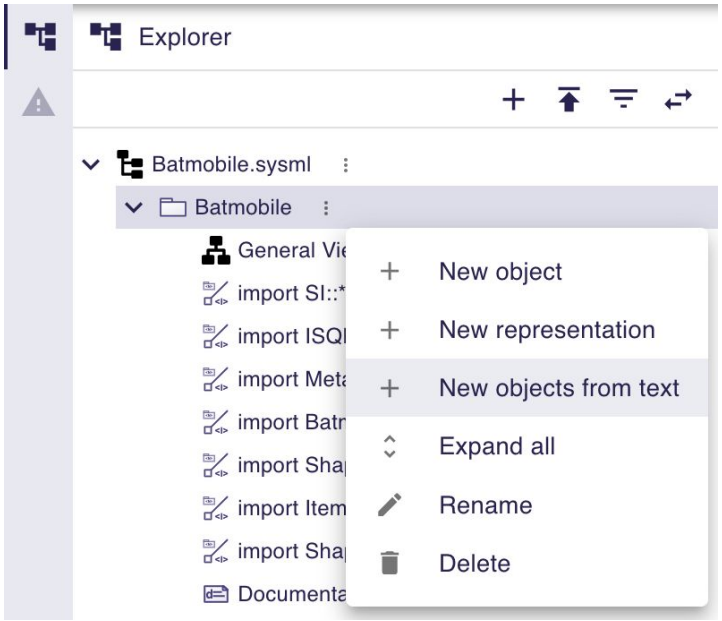
OCX24 OPEN COMMUNITY EXPERIENCE

Sirius Web Extensibility

- Tools



- Menus

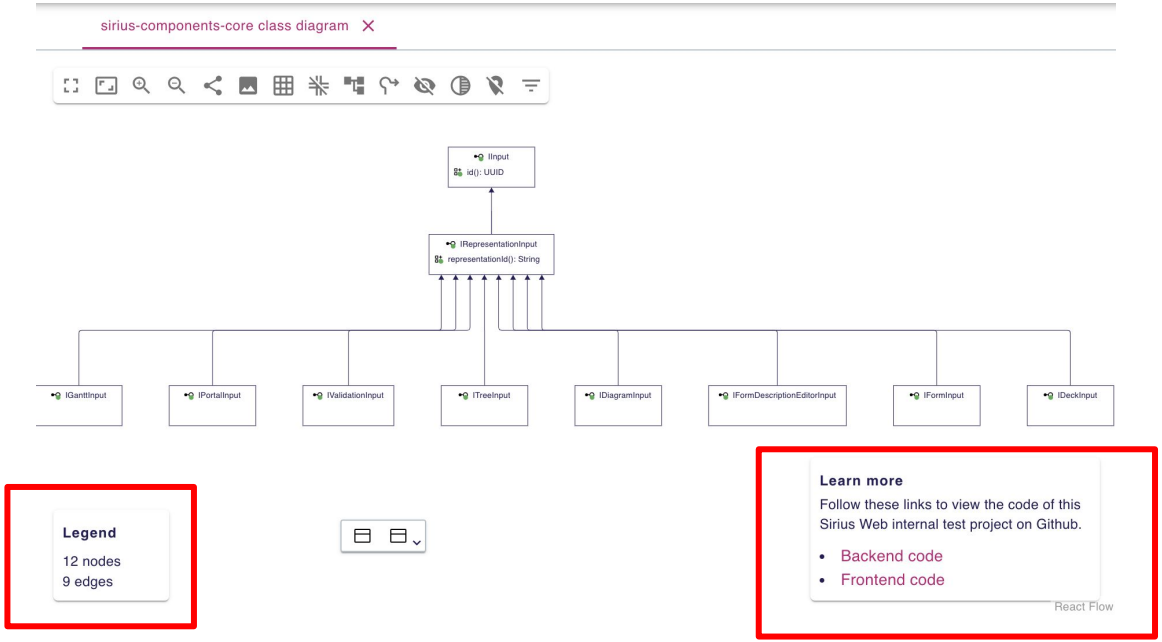


OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

Sirius Web Extensibility

- Diagrams



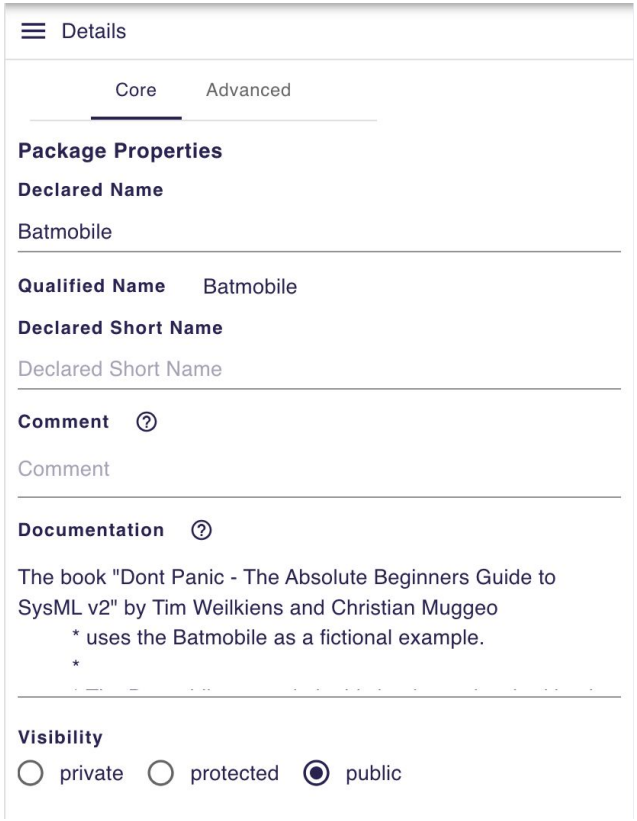
- https://blog.obeosoft.com/sirius-web-2024-9

OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

Sirius Web Customization

- Backend - Details View



Sirius Web Customization

- Backend - Details View

SysMLv2PropertiesConfigurer.java

```java
import org.eclipse.sirius.components.view.form.FormDescription;
import org.eclipse.sirius.components.collaborative.forms.services.api.IPropertiesDescriptionRegistry;
import org.eclipse.sirius.components.collaborative.forms.services.api.IPropertiesDescriptionRegistryConfigurer;

@Configuration
public class SysMLv2PropertiesConfigurer implements IPropertiesDescriptionRegistryConfigurer {
  ...
  @Override
  public void addPropertiesDescriptions(IPropertiesDescriptionRegistry registry) {
      // Build the actual FormDescription that will be used in Detail view.
      FormDescription viewFormDescription = this.createDetailsView();
      ...
      // Convert the View-based FormDescription and register the result into the system
      AQLInterpreter interpreter = new AQLInterpreter(List.of(new DetailsViewService(...), List.of(SysmlPackage.eINSTANCE), ...);
      IRepresentationDescription converted = this.converter.convert(viewFormDescription, List.of(), interpreter);
      if (converted instanceof org.eclipse.sirius.components.forms.description.FormDescription formDescription) {
          formDescription.getPageDescriptions().forEach(registry::add);
      }
  }
}
```

CodeImage

Sirius Web Customization

- Backend - Standard Libraries

OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

Sirius Web Customization

- Backend - Standard Libraries

SysMLEditingContextProcessor.java

```java
import org.eclipse.emf.ecore.resource.ResourceSet;
import org.eclipse.sirius.components.core.api.IEditingContextProcessor;
import org.eclipse.sirius.components.emf.services.api.IEMFEditingContext;
import org.springframework.stereotype.Service;

@Service
public class SysMLEditingContextProcessor implements IEditingContextProcessor {

    private final SysMLStandardLibrariesConfiguration standardLibraries;
    ...
    @Override
    public void preProcess(IEditingContext editingContext) {
        if (editingContext instanceof IEMFEditingContext siriusWebEditingContext) {
            ResourceSet sourceResourceSet = this.standardLibraries.getLibrariesResourceSet();
            ResourceSet targetResourceSet = siriusWebEditingContext.getDomain().getResourceSet();
            // Copy all libraries in the project
            ...
        }
    }

    @Override
    public void postProcess(IEditingContext editingContext) {
    }
}
```

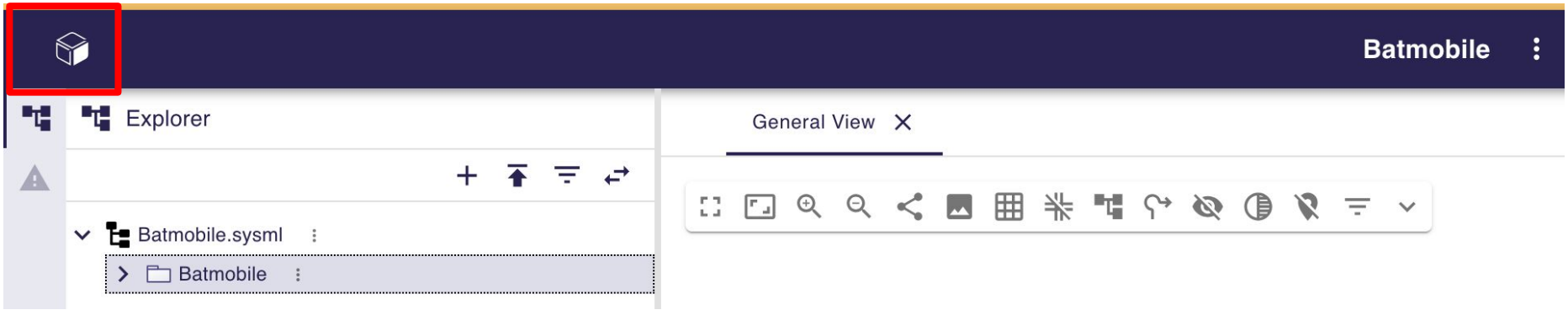
CodeImage

OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

Sirius Web Customization

- Frontend - Icon



Sirius Web Customization

- Frontend - Icon

index.tsx

```tsx
import { ExtensionRegistry } from '@eclipse-sirius/sirius-components-core';
import {
  navigationBarIconExtensionPoint,
  NavigationBarIconProps
} from '@eclipse-sirius/sirius-web-application';

import { SysONIcon } from './core/SysONIcon';

const SysONNavigationBarIcon = ({}: NavigationBarIconProps) => {
  return <SysONIcon />;
};

const extensionRegistry: ExtensionRegistry = new ExtensionRegistry();

extensionRegistry.addComponent(navigationBarIconExtensionPoint, {
  identifier: 'syson_navigationbar#icon',
  Component: SysONNavigationBarIcon,
});
```

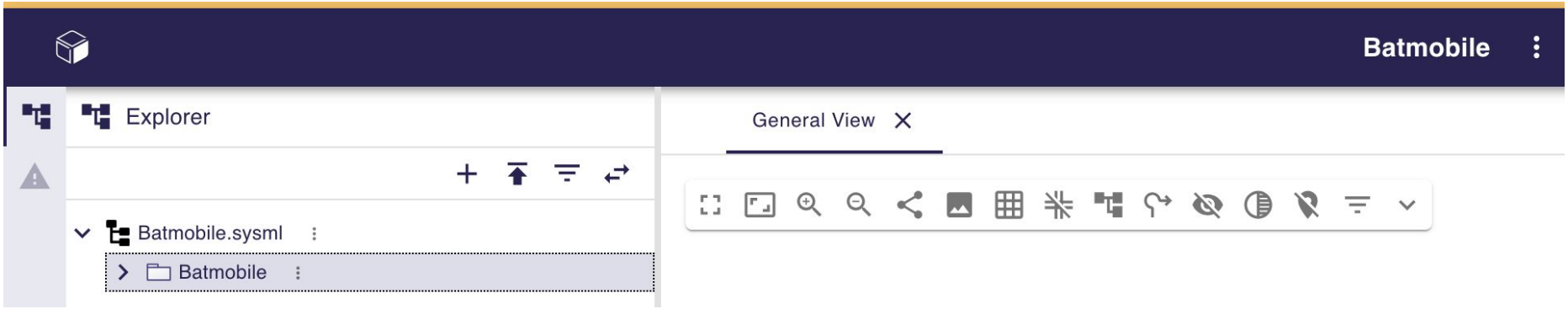
CodeImage

OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

Sirius Web Customization

- Frontend - Theme



Sirius Web Customization

- Frontend - Theme

sysonTheme.ts

```ts
import { theme } from '@eclipse-sirius/sirius-components-core';
import { Theme, createTheme } from '@mui/material/styles';

export const baseTheme: Theme = createTheme({
  ...theme,
  palette: {
    mode: 'light',
    primary: {
      main: '#292253',
      dark: '#64669B',
      light: '#534E75',
    },
  ...
});

export const sysonTheme = createTheme(
  ...
  baseTheme
);
```

CodeImage

index.tsx

```tsx
import {
  SiriusWebApplication,
} from '@eclipse-sirius/sirius-web-application';

import { sysonTheme } from './theme/sysonTheme';

const container = document.getElementById('root');
const root = createRoot(container!);
root.render(
  <SiriusWebApplication
    httpOrigin={httpOrigin}
    wsOrigin={wsOrigin}
    theme={sysonTheme}
    extensionRegistry={extensionRegistry}>
    <DiagramRepresentationConfiguration nodeTypeRegistry={nodeTypeRegistry} />
  </SiriusWebApplication>
);
```

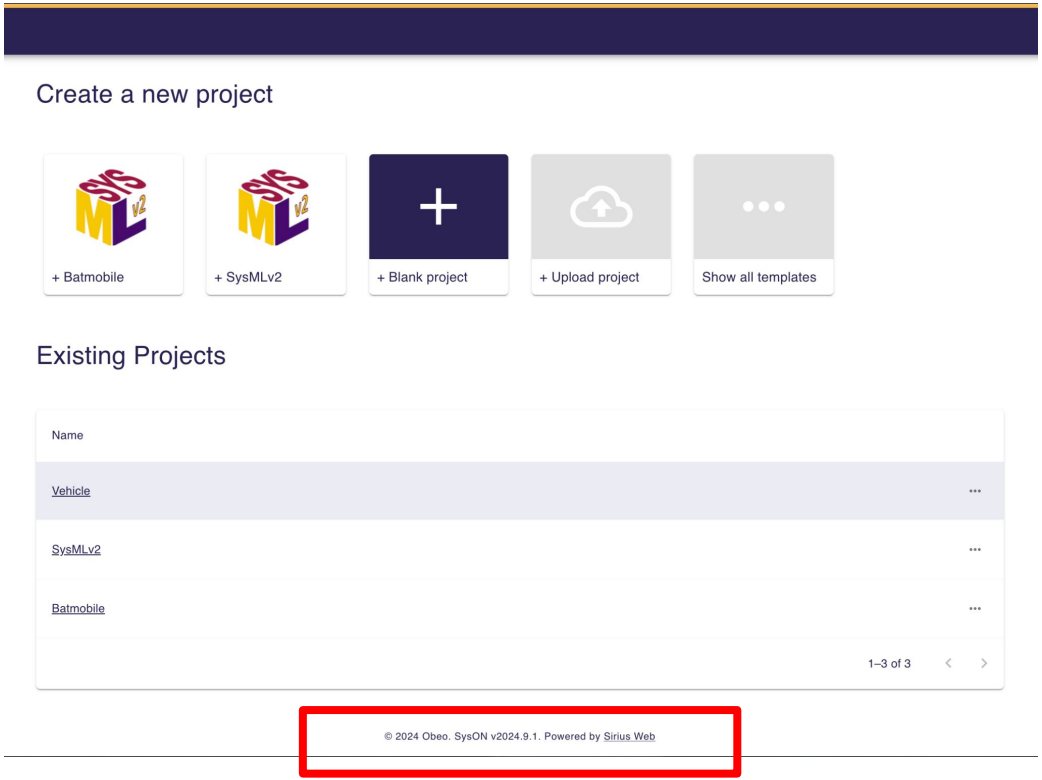
CodeImage

OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

Sirius Web Customization

- Frontend - Footer



Sirius Web Customization

- Frontend - Footer

index.tsx

```tsx
import { ExtensionRegistry } from '@eclipse-sirius/sirius-components-core';
import {
  footerExtensionPoint
} from '@eclipse-sirius/sirius-web-application';

import { SysONFooter } from './extensions/SysONFooter';

const extensionRegistry: ExtensionRegistry = new ExtensionRegistry();

extensionRegistry.addComponent(footerExtensionPoint, {
  identifier: `syson_${footerExtensionPoint.identifier}`,
  Component: SysONFooter,
});
```

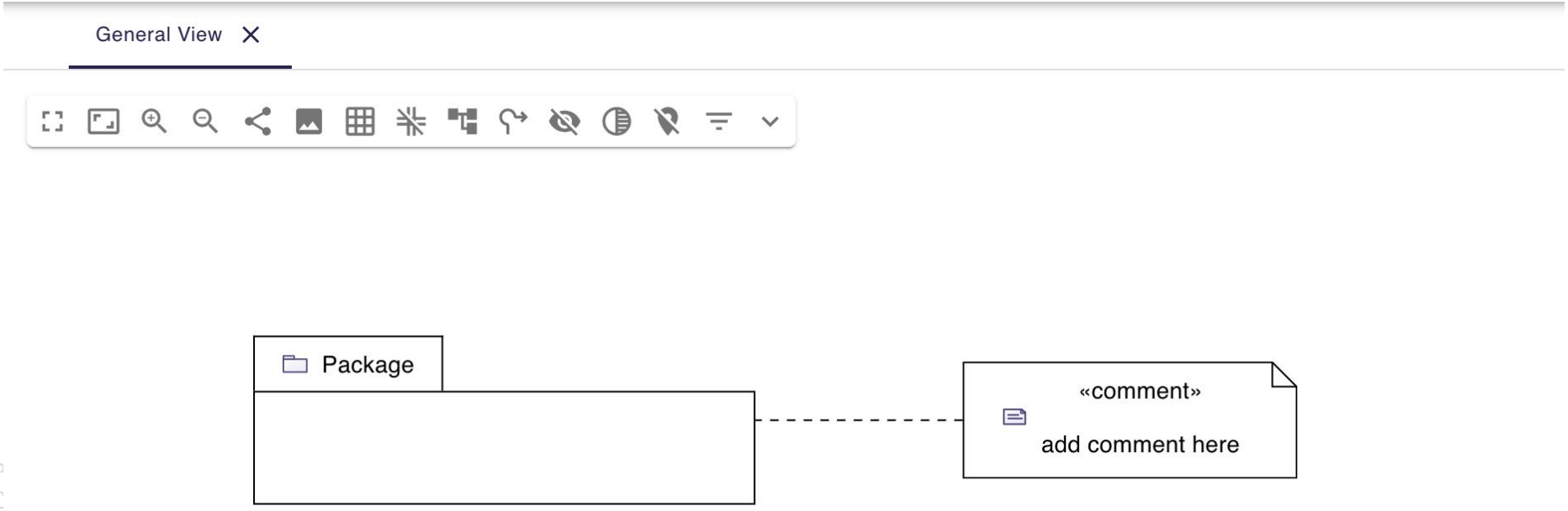
CodeImage

OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

Sirius Web Customization

- Backend/Frontend - Custom Shapes for Diagrams



OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

Sirius Web Customization

- Backend/Frontend - Custom Shapes for Diagrams

index.tsx

```tsx
import {
  DiagramRepresentationConfiguration,
  NodeTypeRegistry,
  SiriusWebApplication,
} from '@eclipse-sirius/sirius-web-application';

const nodeTypeRegistry: NodeTypeRegistry = {
  nodeLayoutHandlers: [new SysMLPackageNodeLayoutHandler(), new SysMLNoteNodeLayoutHandler()],
  nodeConverters: [new SysMLPackageNodeConverter(), new SysMLNoteNodeConverter()],
  nodeTypeContributions: [
    <NodeTypeContribution component={SysMLPackageNode} type={'sysMLPackageNode'} />,
    <NodeTypeContribution component={SysMLNoteNode} type={'sysMLNoteNode'} />,
  ],
};

const container = document.getElementById('root');
const root = createRoot(container!);
root.render(
  <SiriusWebApplication
    httpOrigin={httpOrigin}
    wsOrigin={wsOrigin}
    theme={sysonTheme}
    extensionRegistry={extensionRegistry}>
    <DiagramRepresentationConfiguration nodeTypeRegistry={nodeTypeRegistry} />
  </SiriusWebApplication>
);
```

CodeImage

OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

Sirius Web Customization

- Backend/Frontend - Custom Shapes for Diagrams

SysMLPackageNodeStyleProvider.java

```java
import org.eclipse.sirius.components.diagrams.INodeStyle;
import org.eclipse.sirius.components.view.diagram.NodeStyleDescription;
import org.eclipse.sirius.components.view.emf.diagram.INodeStyleProvider;
import org.eclipse.syson.sysmlcustomnodes.SysMLPackageNodeStyleDescription;

@Service
public class SysMLPackageNodeStyleProvider implements INodeStyleProvider {

    public static final String NODE_SYSML_PACKAGE = "customnode:sysmlpackage";

    @Override
    public Optional<String> getNodeType(NodeStyleDescription nodeStyle) {
        if (nodeStyle instanceof SysMLPackageNodeStyleDescription) {
            return Optional.of(NODE_SYSML_PACKAGE);
        }
        return Optional.empty();
    }

    @Override
    public Optional<INodeStyle> createNodeStyle(NodeStyleDescription nodeStyle, Optional<String> optionalEditingContextId) {
        Optional<INodeStyle> iNodeStyle = Optional.empty();
        Optional<String> nodeType = this.getNodeType(nodeStyle);
        if (nodeType.isPresent()) {
            return Optional.of(SysMLPackageNodeStyle.newSysMLPackageNodeStyle()
                    .background(Optional.ofNullable(((SysMLPackageNodeStyleDescription) nodeStyle).getBackground())
                            .filter(FixedColor.class::isInstance)
                            .map(FixedColor.class::cast)
                            .map(FixedColor::getValue)
                            .orElse("transparent"))
                    .borderColor(Optional.ofNullable(nodeStyle.getBorderColor())
                            .filter(FixedColor.class::isInstance)
                            .map(FixedColor.class::cast)
                            .map(FixedColor::getValue)
                            .orElse("black"))
                    .borderSize(nodeStyle.getBorderSize())
                    .borderStyle(LineStyle.valueOf(nodeStyle.getBorderLineStyle().getLiteral())).build());
        }

        return iNodeStyle;
    }
}
```

CodeImage

OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

Tests in SysON

- Backend
  - JUnit tests
  - Integration tests
- Frontend
  - Cypress

build.yml

```
...
- name: Build the backend
        ...
        run: mvn -U -B -e clean verify --settings settings.xml
...
- name: Run end to end tests against the application
        uses: cypress-io/github-action@v5
        with:
          build: docker compose -f docker-compose-integration-tests.yml up -d
          start: docker compose -f docker-compose-integration-tests.yml ps
          wait-on: "http://localhost:8080/login"
          wait-on-timeout: 180
          working-directory: integration-tests
          record: false
        ...
```

CodeImage

- All backend/frontend tests launched by the CI

Thank you!

- SysON
  - Website: https://mbse-syson.org
  - Github: https://github.com/eclipse-syson/syson
  - Documentation: https://doc.mbse-syson.org
- Sirius Web
  - Website: https://eclipse.dev/sirius/sirius-web.html
- Obeo Cloud Platform
  - Website: https://www.obeosoft.com/en/products/obeo-cloud-platform
- Obeo
  - Website: https://www.obeosoft.com
  - Blog: https://blog.obeosoft.com

OBEO

OCX24 OPEN COMMUNITY EXPERIENCE

Questions?