

TALK

NOTES



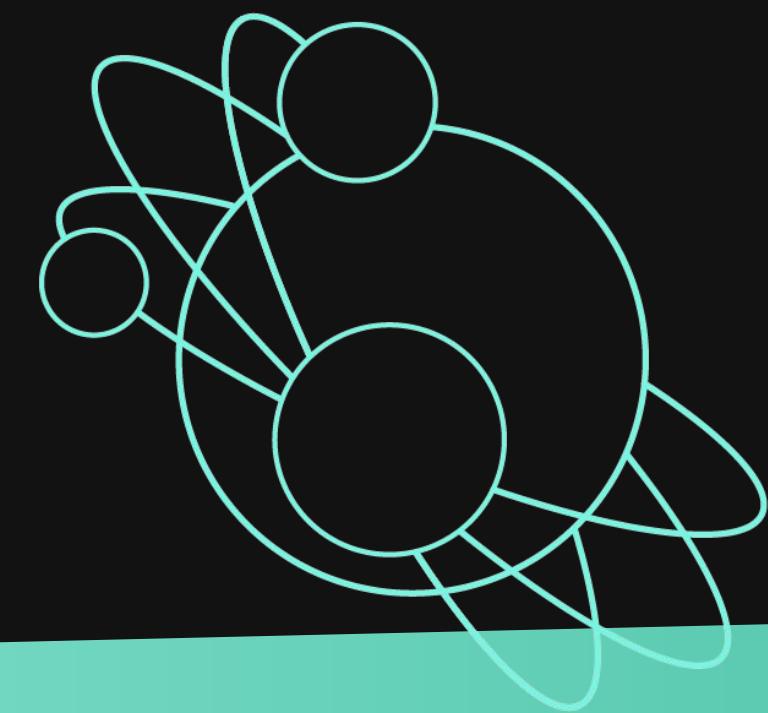


Simplifying Backstage

Breaking the Code Barrier

Ben Lambert & Patrik Oldsberg

INTRO





Ben Lambert



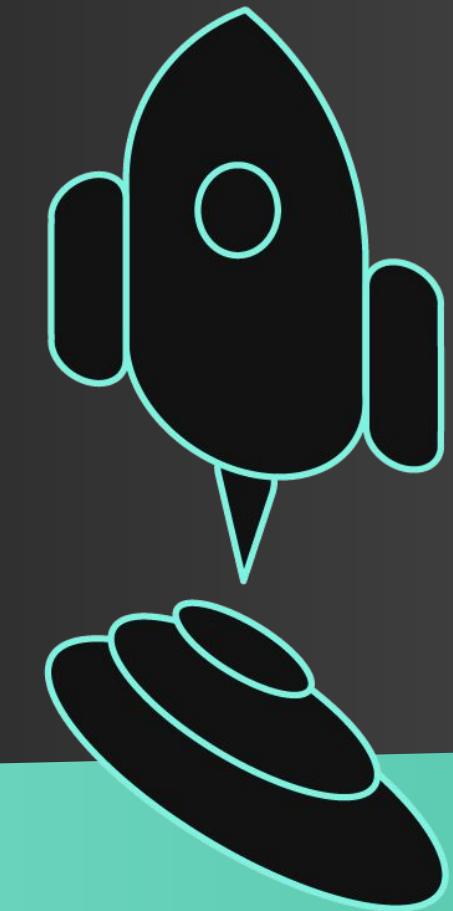
benjdlambert
Senior Engineer @ Spotify
Core Maintainer | Backstage



Patrik Oldsberg

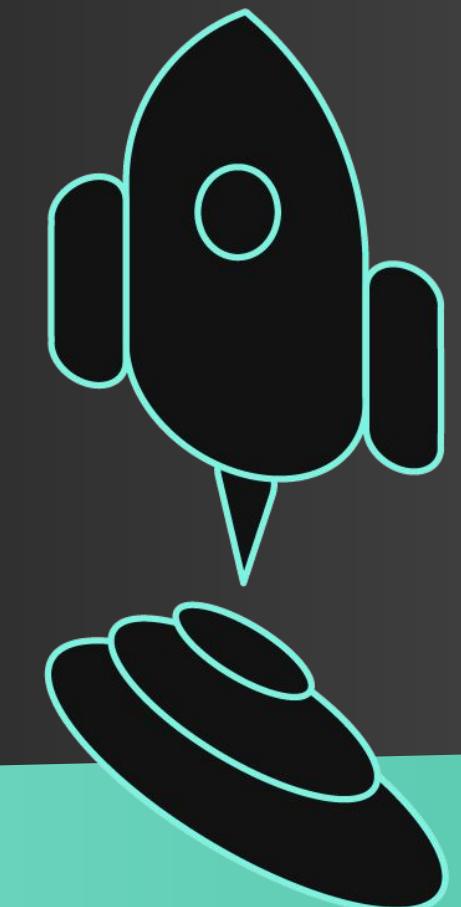


Rugvip
Senior Engineer @ Spotify
Core Maintainer | Backstage

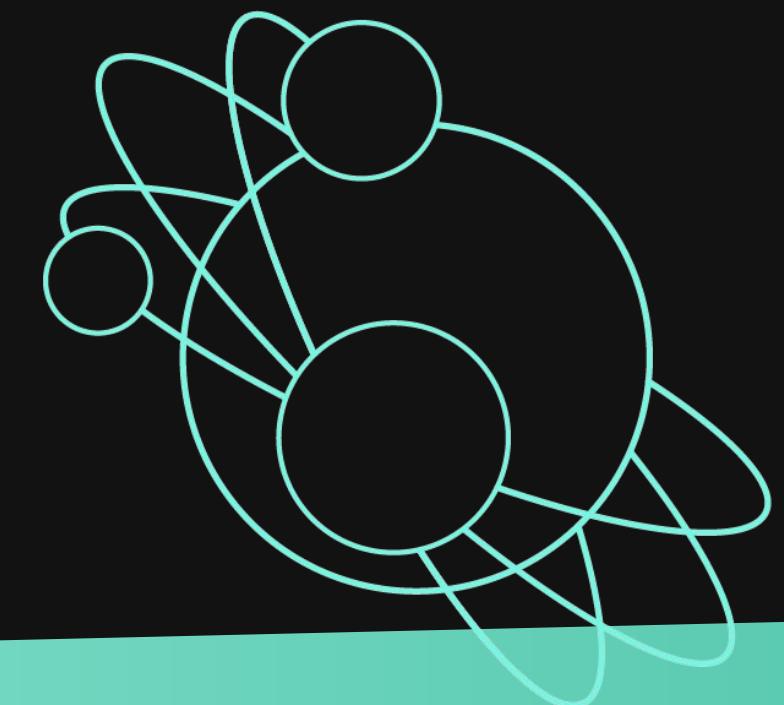


Agenda

- Project Updates
- Frontend Framework Evolution



PROJECT UPDATES





2.3k+

Adopters

↑ 90%

1.2k+

Contributors

↑ 20%

23.9k

Stars

↑ 10%

New Project Areas

Permissions

- Taken on by the Imaginary Goats squad @Spotify

Incubating

Open API

- Aramis Sennyey (@sennyeya)

Incubating

Scaffolder

- Paul Cowan (@dagda1)
- Bogdan Nechyporenko (@acierto)



Backstage ❤️ Plugins

A screenshot of a GitHub repository interface for the 'backstage/backstage' repository. The repository has 412 issues, 92 pull requests, and 174 security vulnerabilities. The main view shows a list of files and sub-directories under the 'backstage/plugins' path. The 'plugins' directory is expanded, showing numerous sub-modules such as 'adr-backend', 'airbrake-backend', 'analytics-module-ga', etc. Each item in the list includes a small icon, the module name, its last commit message (e.g., 'Version Packages (next)'), and the date of the last commit. A search bar at the top right contains the placeholder 'Type [] to search'. The interface is dark-themed.

Name	Last commit message	Last commit date
..		
adr-backend	Version Packages (next)	2 days ago
adr-common	Version Packages	last week
adr	Version Packages (next)	2 days ago
airbrake-backend	Version Packages (next)	2 days ago
airbrake	Version Packages (next)	2 days ago
allure	Version Packages (next)	2 days ago
analytics-module-ga	Version Packages (next)	2 days ago
analytics-module-ga4	Version Packages (next)	2 days ago
analytics-module-newrelic-browser	Version Packages (next)	2 days ago
apache-airflow	Version Packages (next)	2 days ago
api-docs-module-protoc-gen-doc	Version Packages (next)	2 days ago
api-docs	Version Packages (next)	2 days ago
apollo-explorer	Version Packages (next)	2 days ago
app-backend	Version Packages (next)	2 days ago
app-node	Version Packages (next)	2 days ago
auth-backend-module-gcp-iam-provider	Version Packages (next)	2 days ago
auth-backend-module-github-provider	Version Packages (next)	2 days ago
auth-backend-module-gitlab-provider	Version Packages (next)	2 days ago
auth-backend-module-google-provider	Version Packages (next)	2 days ago
auth-backend-module-microsoft-provider	mistakenly changed secret marker	yesterday
auth-backend-module-oauth2-provider	Version Packages (next)	2 days ago
auth-backend-module-pinned-provider	Version Packages (next)	2 days ago
auth-backend	Version Packages (next)	2 days ago
auth-node	Version Packages (next)	2 days ago

Free but with a cost

- Monorepo was good for breaking changes early on
- Many plugins without owner, falls back on core maintainers
- More load on Maintainers



Step 1.

The screenshot shows a GitHub issue page for the repository `backstage/backstage`. The issue is titled "RFC: Stop accepting new plugins to the backstage/backstage repository" and has the identifier #20266. The status is "Open" and there are 2 tasks done. The issue was opened by `Rugvip` last month, with 18 comments. The main content of the issue discusses the rationale behind stopping plugin contributions to the main repository, the proposal to stop accepting new plugins, the development plan, and the risks involved. The sidebar on the right provides options for managing the issue, including assignees, labels, projects, milestones, notifications, and participants.

RFC: Stop accepting new plugins to the backstage/backstage repository
#20266

`Rugvip` commented last month

Need

Starting out the Backstage project we thought it was important to have a large number of plugins available in the main repository. Both in order to help us make more informed decisions when building out the core framework and features, and also to help keep the project healthy and active with a lot of communication centered in one place.

The project has now grown (a lot! 🎉), and we have plenty of plugins to keep the project active and help guide the evolution of the framework. A little bit too many in fact. We've had previous calls for contributions like [#8294](#), but with the core maintainers being the owners of any plugin without a [Project Area](#) or [Plugin Maintainer](#), it remains to be a significant workload for the core maintainers team. We want to reduce this workload, both to free up time for improving the project in other ways, but also to make sure that broadly used plugins receive the care that they deserve.

Proposal

We will stop accepting new plugins to the `backstage/backstage` repository. Exceptions are made for existing project areas that want to expand their set of plugins, as well as new library packages for existing plugins.

We recognize that there is some work involved in setting up a repository to publish plugins from. We will provide a GitHub template repo with a base setup to help out. Plugins will not be able to use the `@backstage` namespace on NPM. They should instead use the `@<org>/backstage-plugin-*` or `backstage-plugin-*` naming pattern. Plugin authors can keep adding their plugins to <https://backstage.io/plugins> to make them easier to discover.

This is a step towards ensuring that the Backstage has a growing and healthy ecosystem of plugins, and we will keep working towards this in other ways. If you are a plugin maintainer that would like to help out by moving your plugin out of the `backstage/backstage` repo, please let us know and we can help get that sorted. There will be more to come on this in the future, so this is just an early opportunity for those alright with a slightly rockier road as we figure things out.

Alternatives

We have many times discussed whether we should have a "community plugins" or "plugins contrib" repository in the Backstage GitHub organization. In the end we feel that it would encounter the same issues as what we're seeing in the main repository right now, so it is something we'd like to avoid. The Backstage Open Source project is from the very beginning built towards enabling an open and distributed ecosystem of plugins, with limited need for centralized control.

Risks

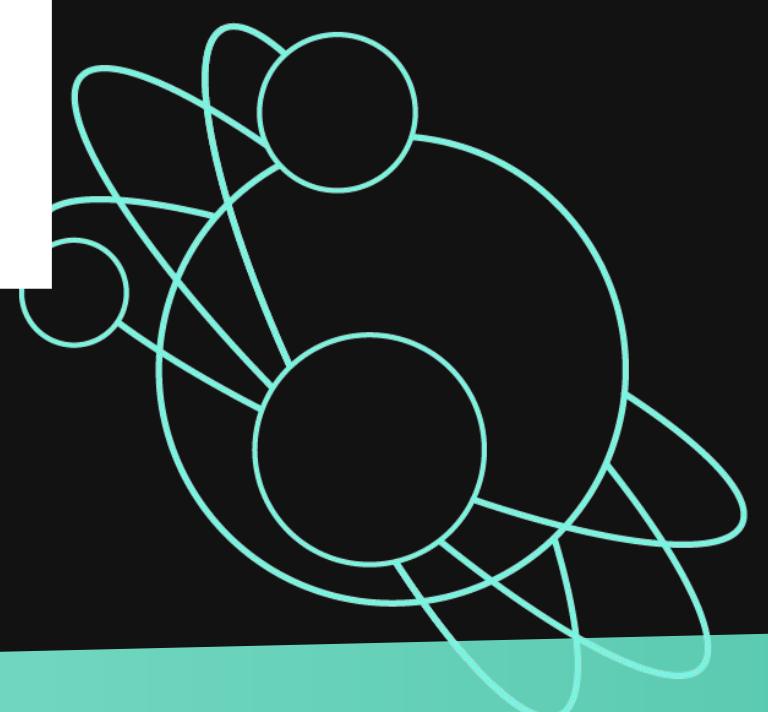
There are some benefits to be had to have all plugins in one repository when it comes to keeping plugins up to date, as you can make broader changes more easily. At the same time it also adds friction since you end up in situations where you are forced to make upgrades early. The Backstage framework is in many ways designed to allow for plugins to not always be on the latest and greatest version of the core APIs. It is fine for plugins to lag behind and only do upgrades when necessary or to access new features. This means that while broad upgrades can be slightly more effective, more narrow ones can often be skipped altogether. There is of course also the fact that we are looking to distribute this workload, it is the reason for this RFC.

Another risk is that we simply do not have enough plugins in the main repository to properly test new features that we are building in the core framework. This is not a problem we have right now, but in the future we may need further investment.

Backstage ❤ Plugins

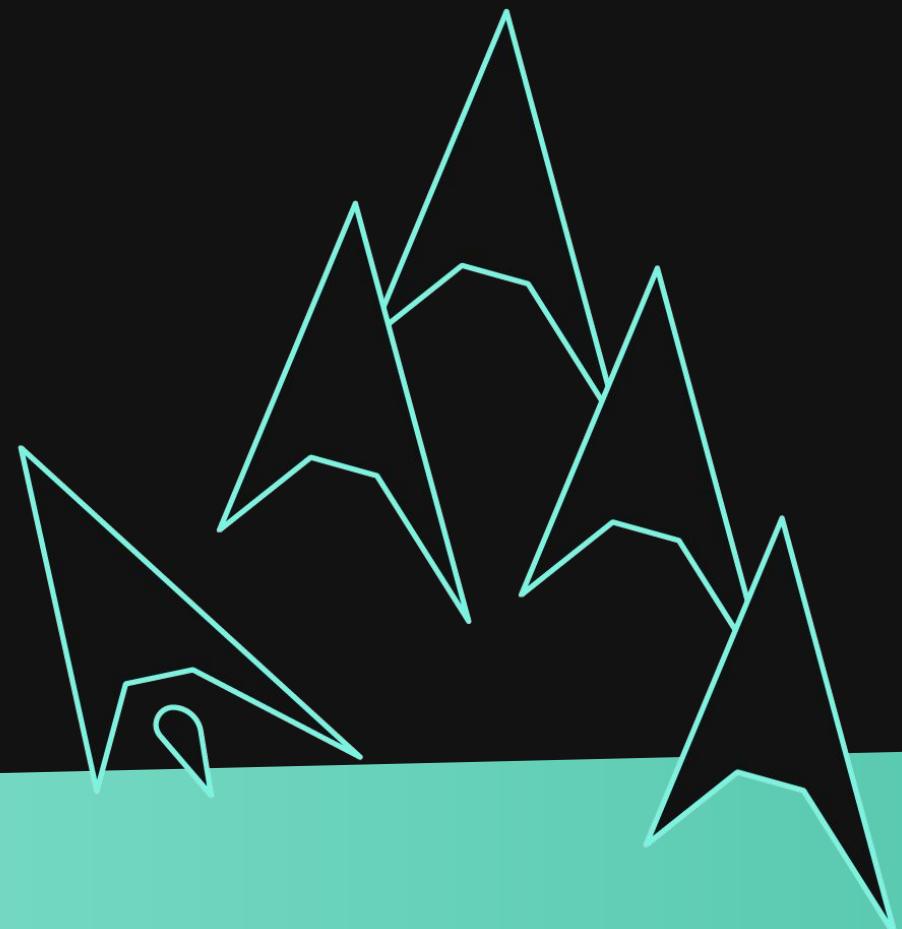
- PagerDuty
 - <https://github.com/backstage/backstage/issues/19347>
- CircleCI
 - <https://github.com/CircleCI-Public/backstage-plugin/issues/3>
- Yours?

FRAMEWORK EVOLUTION



Frontend Framework Evolution

Breaking the Code Barrier



Integration with Code



```
import { GraphQLPage } from '@backstage/plugin-graphiql';

const routes = (
  <FlatRoutes>
    {/* Add the line below */}
    <Route path="/graphiql" element={<GraphQLPage />} />
```

```
const customEntityFilterKind = ['Component', 'API', 'System'];

const EntityLayoutWrapper = (props: { children?: ReactNode }) => {
  const [badgesDialogOpen, setBadgesDialogOpen] = useState(false);
  const [playlistDialogOpen, setPlaylistDialogOpen] = useState(false);

  const extraMenuItems = useMemo(() => {
    return [
      {
        title: 'Badges',
        Icon: BadgeIcon,
        onClick: () => setBadgesDialogOpen(true),
      },
      {
        title: 'Add to playlist',
        Icon: PlaylistAddIcon,
        onClick: () => setPlaylistDialogOpen(true),
      },
    ];
  }, []);

  return (
    <EntityLayout
      UNSTABLE_extraContextMenuItems={extraMenuItems}
      UNSTABLE_contextMenuOptions={{ disableUnregister: 'visible' }}
    >
      {props.children}
    </EntityLayout>
    <EntityBadgesDialog
      open={badgesDialogOpen}
      onClose={() => setBadgesDialogOpen(false)}
    />
    <EntityPlaylistDialog
      open={playlistDialogOpen}
      onClose={() => setPlaylistDialogOpen(false)}
    />
  );
}

const techdocsContent = (
  <EntityTechdocsContent>
    <TechDocAddons>
      <TextSize />
      <ReportIssue />
      <LightBox />
    </TechDocAddons>
  </EntityTechdocsContent>
);

export const ciCdContent = (
  <EntitySwitch>
    <EntitySwitch.Case if={isJenkinsAvailable}>
      <EntityJenkinsContent />
    </EntitySwitch.Case>
    <EntitySwitch>
      <EntitySwitch.Case if={isAzureDevOpsAvailable}>
        <EntityGitHubActionsCard />
        <EntityPipelineCard />
        <EntityPipelinesAvailable>
          <Text defaultLimit={25} />
        </EntityPipelinesAvailable>
        <EntityOctopusDeployAvailable>
          <Text defaultLimit={25} />
        </EntityOctopusDeployAvailable>
      </EntitySwitch.Case>
      <EntitySwitch.Case if={isTravisCIAvailable}>
        <EntityTravisCIContent />
      </EntitySwitch.Case>
      <EntitySwitch.Case if={isGoCDAvailable}>
        <EntityGoCDContent />
      </EntitySwitch.Case>
    </EntitySwitch>
    <EntitySwitch.Case if={isBuildkiteAvailable}>
      <EntityBuildkiteContent />
    </EntitySwitch.Case>
    <EntitySwitch.Case if={isCircleCIAvailable}>
      <EntityCircleCIContent />
    </EntitySwitch.Case>
    <EntitySwitch.Case if={isCloudbuildAvailable}>
      <EntityCloudbuildContent />
    </EntitySwitch.Case>
    <EntitySwitch.Case if={isTravisCIAvailable}>
      <EntityTravisCIContent />
    </EntitySwitch.Case>
    <EntitySwitch.Case if={isLinguiAvailable}>
      <EntityLinguiContent />
    </EntitySwitch.Case>
    <EntitySwitch.Case if={isNomadJobAvailable}>
      <Grid item sm={6}>
        <EntityNomadJobOverviewCard />
      </Grid>
    </EntitySwitch.Case>
    <EntitySwitch.Case if={isGithubActionsAvailable}>
      <Grid item sm={6}>
        <EntityRecentGithubActionsRunsCard limit={4} variant="gridItem" />
      </Grid>
    </EntitySwitch.Case>
    <EntitySwitch.Case if={isOrphan}>
      <Grid item xs={12}>
        <EntityOrphanWarning />
      </Grid>
    </EntitySwitch.Case>
    <EntitySwitch.Case if={hasRelationWarnings}>
      <Grid item xs={12}>
        <EntityRelationWarning />
      </Grid>
    </EntitySwitch.Case>
    <EntitySwitch.Case if={hasCatalogProcessingErrors}>
      <Grid item xs={12}>
        <EntityProcessingErrorsPanel />
      </Grid>
    </EntitySwitch.Case>
  </EntitySwitch>
  <EntityWarningContent />
  <EntitySwitch.Case if={isOrphan}>
    <Grid item xs={12}>
      <EntityOrphanWarning />
    </Grid>
  </EntitySwitch.Case>
  <EntitySwitch.Case if={hasRelationWarnings}>
    <Grid item xs={12}>
      <EntityRelationWarning />
    </Grid>
  </EntitySwitch.Case>
  <EntitySwitch.Case if={hasCatalogProcessingErrors}>
    <Grid item xs={12}>
      <EntityProcessingErrorsPanel />
    </Grid>
  </EntitySwitch.Case>

```



```
</Grid>
<EntitySwitch.Case if={isRollbarAvailable}>
  <EntityRollbarContent />
</EntitySwitch.Case>
</EntityLayout.Route>

<EntitySwitch.Case>
  <EntitySentryContent />
</EntitySwitch.Case>
</EntityLayout.Route>

<EntityLayout.Route path="/feedback" title="Feedback">
  <EntityFeedbackResponseContent />
</EntityLayout.Route>
</EntityLayout.Wrapper>

const pullRequestsContent = (
  <EntitySwitch>
    <EntitySwitch.Case if={isAzureDevOpsAvailable}>
      <EntityAzurePullRequestsContent defaultLimit={25} />
    </EntitySwitch.Case>
    <EntitySwitch.Case>
      <EntityGitHubPullRequestsContent />
    </EntitySwitch.Case>
  </EntitySwitch>
);

const overviewContent = (
  <Grid container spacing={3} alignItems="stretch">
    <EntityWarningContent />
    <Grid item md={6} xs={12}>
      <EntityAboutCard variant="gridItem" />
    </Grid>
    <Grid item md={6} xs={12}>
      <EntityCatalogGraphCard variant="gridItem" height={400} />
    </Grid>
  </Grid>
);

const pullRequestsContent = (
  <EntitySwitch>
    <EntitySwitch.Case if={isPagerDutyAvailable}>
      <EntityPagerDutyCard />
    </EntitySwitch.Case>
    <EntitySwitch.Case>
      <EntityNomadAllocationListTable />
    </EntitySwitch.Case>
  </EntitySwitch>
);

const componentPage = (
  <EntitySwitch>
    <EntitySwitch.Case if={isComponentType('service')}>
      <EntityEntityPage />
    </EntitySwitch.Case>
    <EntitySwitch.Case if={isComponentType('website')}>
      <EntityGitHubInsightsContent />
    </EntitySwitch.Case>
    <EntitySwitch.Case>
      <EntitySwitch.Case if={defaultEntityPage}>
        <EntitySwitch.Case if={isScorecardsAvailable}>
          <EntityTechInsightsScorecardCard />
          <Grid item xs={12} md={6}>
            <EntityTechInsightsScorecardCard title="Scorecard 1" description="This is a sample scorecard no. 1" checksId={[{titleCheck}]} />
            <Grid item xs={12} md={6}>
              <EntityTechInsightsScorecardCard title="Scorecard 2" checksId={[{techDocsCheck}]} />
            </Grid>
          </Grid>
        </EntitySwitch.Case>
        <EntityLayout.Route path="/code-coverage" title="Code Coverage">
          <EntityCodeCoverageContent />
        </EntityLayout.Route>
      </EntitySwitch.Case>
    </EntitySwitch.Case>
  </EntitySwitch>
);

const apiPage = (
  <EntityLayoutWrapper>
    <EntityLayout.Route path="/" title="Overview">
      <Grid container spacing={3}>
        <EntityWarningContent />
        <Grid item md={6} xs={12}>
          <EntityAboutCard variant="gridItem" />
        </Grid>
        <Grid item md={6} xs={12}>
          <EntityCatalogGraphCard variant="gridItem" height={400} />
        </Grid>
        <Grid item md={6}>
          <EntityHasSystemsCard variant="gridItem" />
        </Grid>
        <Grid item md={2}>
          <InfoCard title="Rate this entity" />
        </Grid>
        <Grid item md={4}>
          <LikeDislikeButtons />
        </Grid>
      </Grid>
    </EntityLayout.Route>
    <EntityLayout.Route path="/feedback" title="Feedback">
      <EntityFeedbackResponseContent />
    </EntityLayout.Route>
  </EntityLayoutWrapper>
);

const domainPage = (
  <EntityLayoutWrapper>
    <EntityLayout.Route path="/" title="Overview">
      <Grid container spacing={3} alignItems="stretch">
        <EntityWarningContent />
        <Grid item md={6}>
          <EntityAboutCard variant="gridItem" />
        </Grid>
        <Grid item md={6} xs={12}>
          <EntityCatalogGraphCard variant="gridItem" height={400} />
        </Grid>
        <Grid item md={6}>
          <EntityHasSystemsCard variant="gridItem" />
        </Grid>
        <Grid item md={2}>
          <InfoCard title="Rate this entity" />
        </Grid>
        <Grid item md={4}>
          <LikeDislikeButtons />
        </Grid>
      </Grid>
    </EntityLayout.Route>
    <EntityLayout.Route path="/feedback" title="Feedback">
      <EntityFeedbackResponseContent />
    </EntityLayout.Route>
  </EntityLayoutWrapper>
);

const groupPage = (
  <EntityLayoutWrapper>
    <EntityLayout.Route path="/" title="Overview">
      <Grid container spacing={3}>
        <EntityWarningContent />
        <Grid item xs={12}>
          <EntityGroupProfileCard variant="gridItem" />
        </Grid>
      </Grid>
    </EntityLayout.Route>
    <EntityLayout.Route path="/dependencies" title="Dependencies">
      <EntityDependsOnComponentsCard variant="gridItem" />
    </EntityLayout.Route>
    <EntityLayout.Route path="/errors" title="Errors">
      <EntityDependsOnResourcesCard variant="gridItem" />
    </EntityLayout.Route>
  </EntityLayoutWrapper>
);

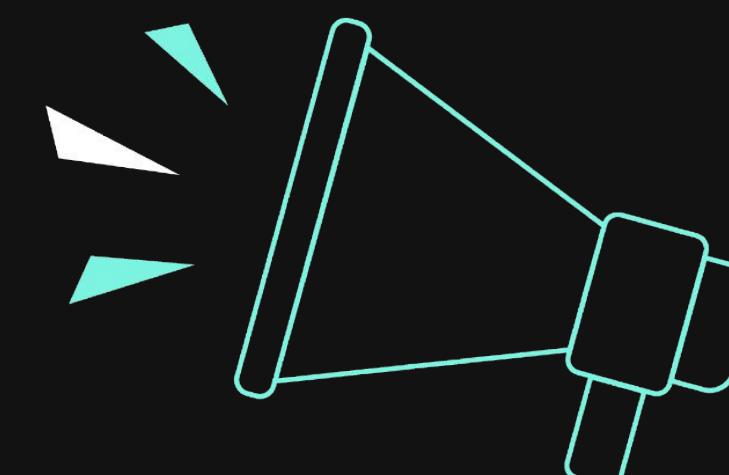
const groupPage = (
  <EntityLayoutWrapper>
    <EntityLayout.Route path="/" title="Overview">
      <Grid container spacing={3}>
        <EntityWarningContent />
        <Grid item xs={12}>
          <EntityGroupProfileCard variant="gridItem" />
        </Grid>
      </Grid>
    </EntityLayout.Route>
    <EntityLayout.Route path="/docs" title="Docs">
      <EntityGroupContent />
    </EntityLayout.Route>
    <EntityLayout.Route path="/kubernetes" title="Kubernetes">
      <EntityKubernetesContent />
    </EntityLayout.Route>
    <EntityLayout.Route path="/dynatrace" title="Dynatrace">
      <EntityDynatraceAvailable>
        <EntityDynaTraceTab />
      </EntityDynatraceAvailable>
    </EntityLayout.Route>
    <EntityLayout.Route path="/git-tags" title="Git Tags">
      <EntityGitTagsAvailable>
        <EntityAzureGitTagsContent />
      </EntityGitTagsAvailable>
    </EntityLayout.Route>
    <EntityLayout.Route path="/pull-requests" title="Pull Requests">
      <EntityPullRequestsContent />
    </EntityLayout.Route>
    <EntityLayout.Route path="/code-insights" title="Code Insights">
      <EntityCodeInsightsContent />
    </EntityLayout.Route>
    <EntityLayout.Route path="/code-coverage" title="Code Coverage">
      <EntityCodeCoverageContent />
    </EntityLayout.Route>
  </EntityLayoutWrapper>
);

const systemPage = (
  <EntityLayoutWrapper>
    <EntityLayout.Route path="/" title="Overview">
      <Grid container spacing={3} alignItems="stretch">
        <EntityWarningContent />
        <Grid item md={6}>
          <EntityAboutCard variant="gridItem" />
        </Grid>
        <Grid item md={6} xs={12}>
          <EntityCatalogGraphCard variant="gridItem" height={400} />
        </Grid>
        <Grid item md={6}>
          <EntityHasComponentsCard variant="gridItem" />
        </Grid>
        <Grid item md={6}>
          <EntityHasApisCard variant="gridItem" />
        </Grid>
        <Grid item md={6}>
          <EntityHasResourcesCard variant="gridItem" />
        </Grid>
      </Grid>
    </EntityLayout.Route>
    <EntityLayout.Route path="/feedback" title="Feedback">
      <EntityFeedbackResponseContent />
    </EntityLayout.Route>
  </EntityLayoutWrapper>
);

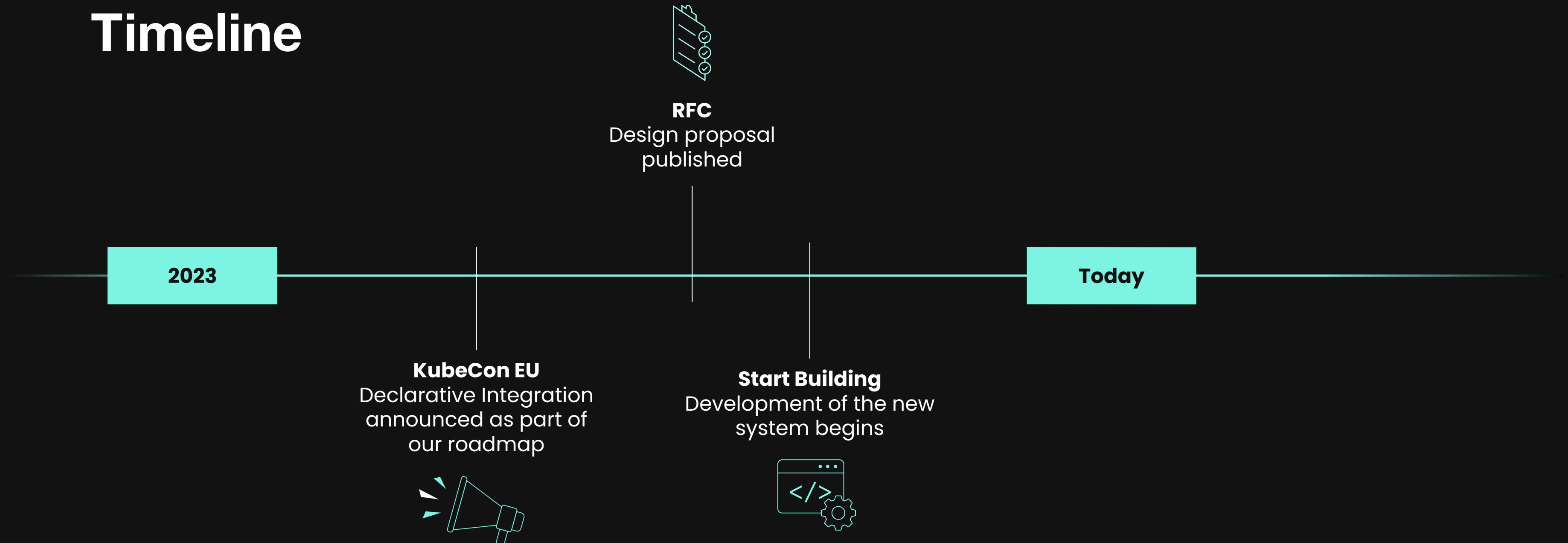
export const entityPage = (
  <EntitySwitch>
    <EntitySwitch.Case if={isKind('component')}>
      <EntityAPICard />
    </EntitySwitch.Case>
    <EntitySwitch.Case if={isKind('group')}>
      <EntityGroupProfileCard variant="gridItem" />
    </EntitySwitch.Case>
    <EntitySwitch.Case if={isKind('user')}>
      <EntityUserCard />
    </EntitySwitch.Case>
    <EntitySwitch.Case if={isKind('system')}>
      <EntitySystemCard />
    </EntitySwitch.Case>
    <EntitySwitch.Case if={isKind('domain')}>
      <EntityDomainCard variant="gridItem" />
    </EntitySwitch.Case>
    <EntitySwitch.Case if={isKind('resource')}>
      <EntityResourceCard />
    </EntitySwitch.Case>
    <EntitySwitch.Case if={defaultEntityPage}>
      <EntitySwitch.Case if={isScorecardsAvailable}>
        <EntityScorecardsContent />
      </EntitySwitch.Case>
    </EntitySwitch.Case>
  </EntitySwitch>
);
```

Introducing...

Declarative Integration



Timeline



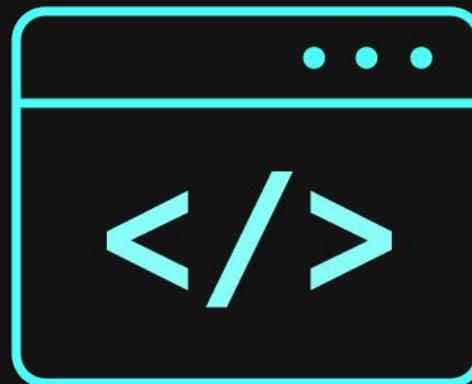
Design Goals

Integration without
TypeScript

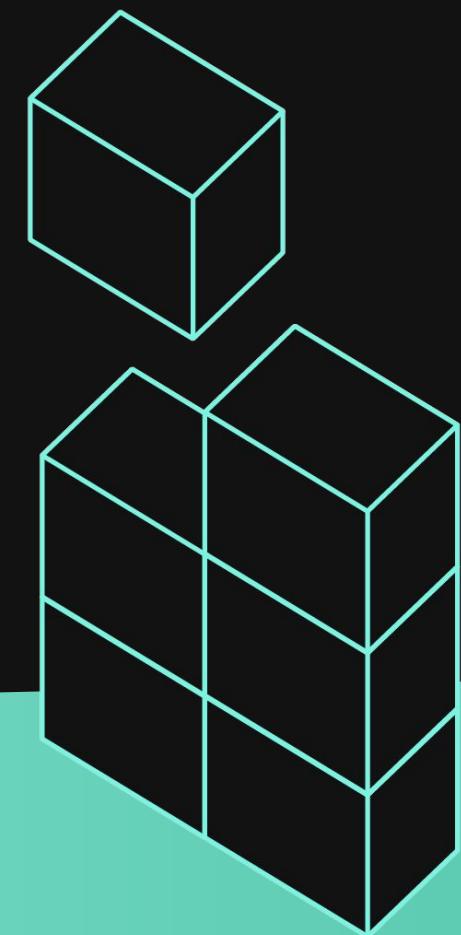
Cleaner composition
patterns

Plugins provide the
defaults

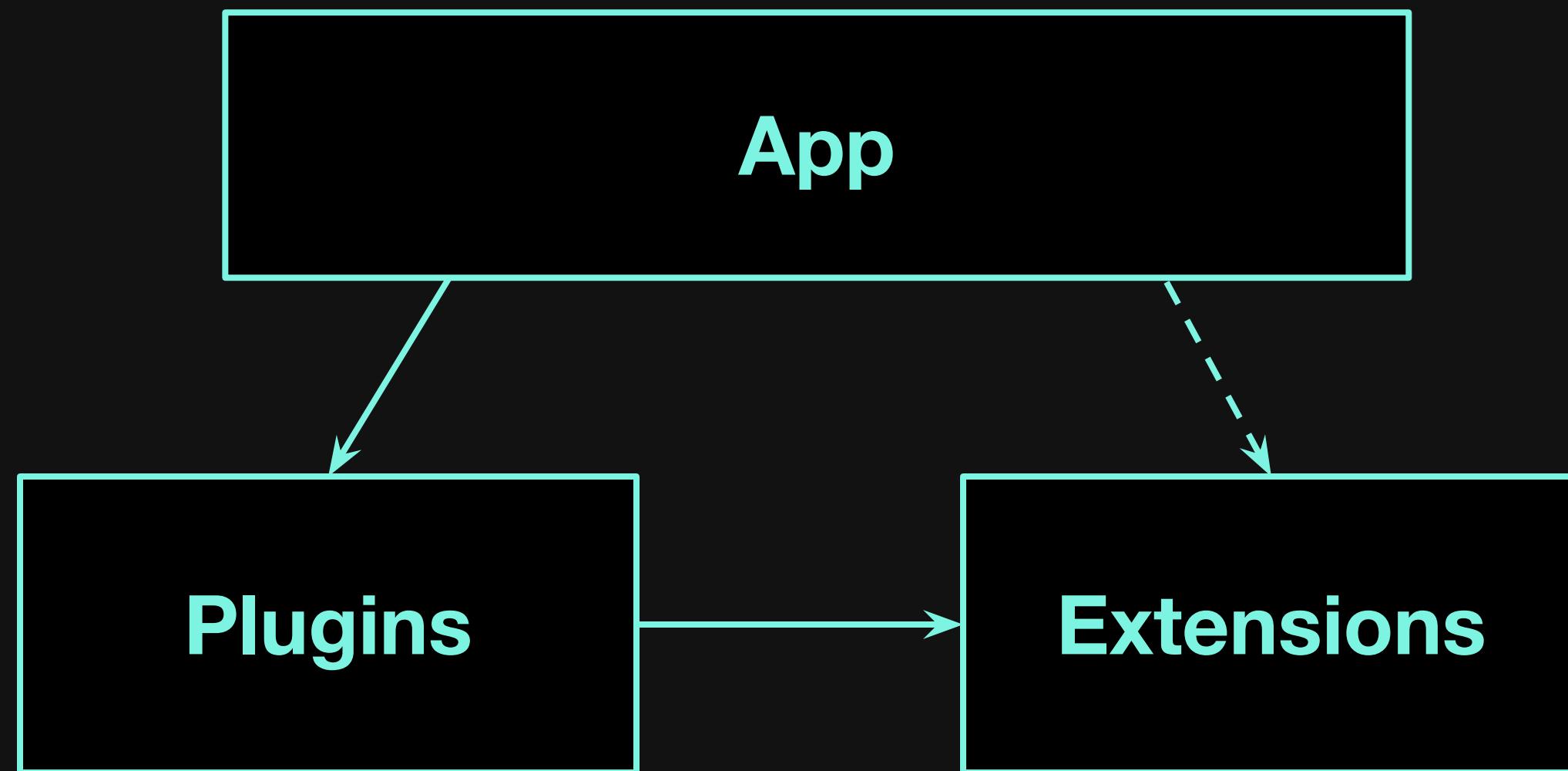
Pave the way for dynamic
plugin installation



Design



Architecture



Extensions



```
import { GraphQLPage } from '@backstage/plugin-graphiql';

const routes = (
  <FlatRoutes>
    {/* Add the line below */}
    <Route path="/graphiql" element={<GraphQLPage />} />
```

Extensions

- ... provide their own attachment point

```
import { GraphQLPage } from '@backstage/plugin-graphiql';

const routes = (
  <FlatRoutes>
    {/* Add the line below */}
    <Route path="/graphiql" element={<GraphQLPage />} />
```

Extensions

- ... provide their own attachment point
- ... can provide a default integration configuration



```
import { GraphQLPage } from '@backstage/plugin-graphiql';

<Route path="/graphiql" element={<GraphQLPage />} />
```

Extensions

- ... provide their own attachment point
- ... can provide a default integration configuration
- ... are not composed using React



```
import { GraphQLPage } from '@backstage/plugin-graphiql';
```

```
<Route
```

```
      element={<GraphQLPage />} />
```

Extensions

- ... provide their own attachment point
- ... can provide a default integration configuration
- ... are not composed using React
- ... are imported automatically



```
import { GraphQLPage } from '@backstage/plugin-graphiql';
```

Extensions

- ... provide their own attachment point
- ... can provide a default integration configuration
- ... are not composed using React
- ... are imported automatically



App Structure - Old

```
● ● ●

const routes = (
  <FlatRoutes>
    <Route path="/graphql" element={<GraphQLPage />} />

    <Route path="/settings" element={<UserSettingsPage />}>
      <SettingsLayout.Route path="/advanced" title="Advanced">
        <AdvancedSettings />
      </SettingsLayout.Route>
    </Route>
  </FlatRoutes>
)

app.createRoot(
  <AppRouter>
    <Root>
      {routes}
    </Root>
  </AppRouter>
)
```

App Structure - Comparison

```
const routes = (
  <FlatRoutes>
    <Route path="/graphql" element={<GraphQLPage />} />

    <Route path="/settings" element={<UserSettingsPage />}>
      <SettingsLayout.Route path="/advanced" title="Advanced">
        <AdvancedSettings />
      </SettingsLayout.Route>
    </Route>
  </FlatRoutes>
)

app.createRoot(
  <AppRouter>
    <Root>
      {routes}
    </Root>
  </AppRouter>
)
```

```
app:
  extensions:
    - 'app'
    - 'app.layout':
      attachTo:
        id: 'app'
        input: root

    - 'app.routes':
      attachTo:
        id: 'app.layout'
        input: content

    - 'plugin.graphql.page':
      attachTo:
        id: 'app.routes'
        input: routes
      config:
        path: /graphql

    - 'plugin.settings.page':
      attachTo:
        id: 'app.routes'
        input: routes
      config:
        path: /settings

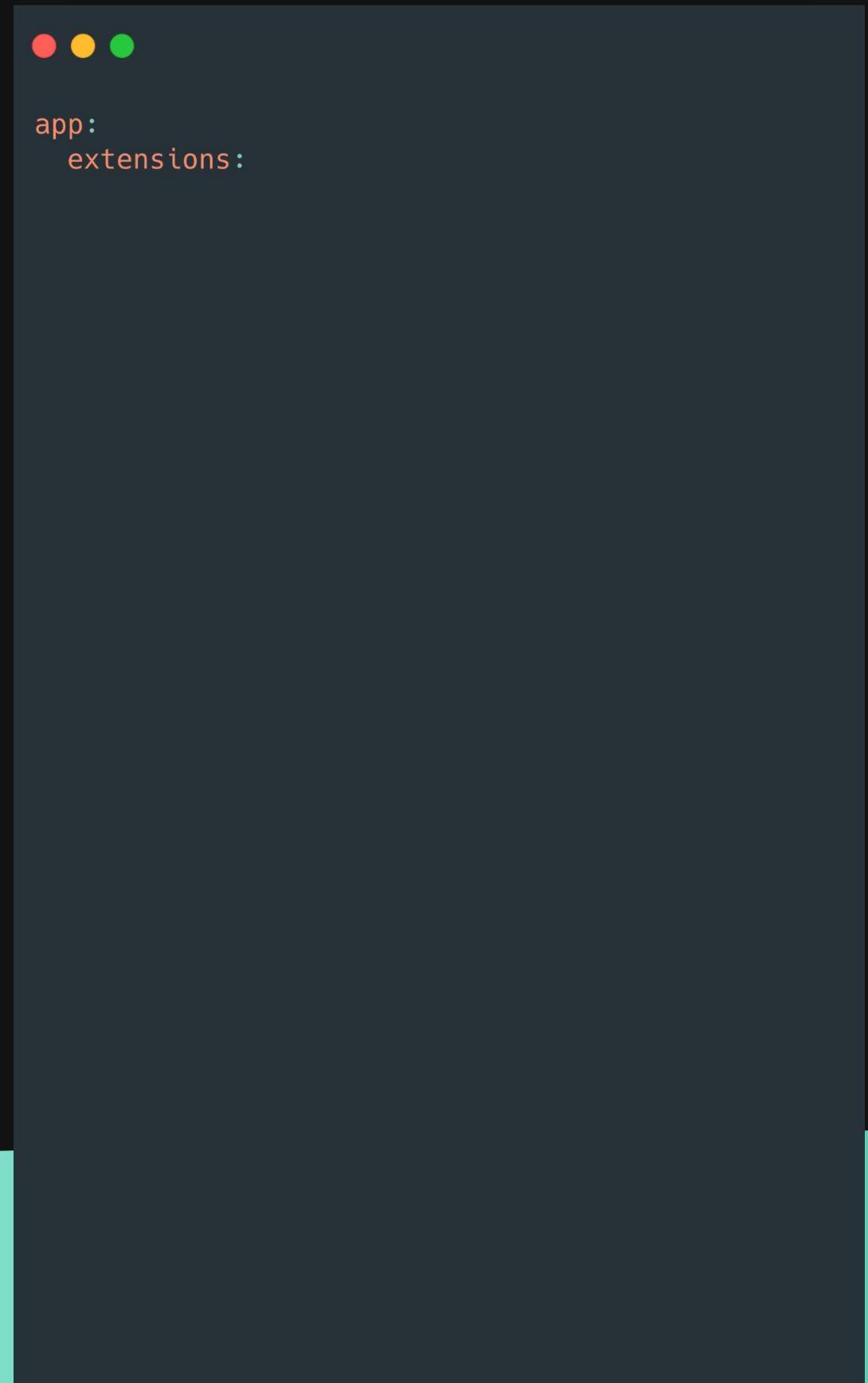
    - 'plugin.settings.tabs.advanced':
      attachTo:
        id: 'plugin.settings.page'
        input: tabs
      config:
        path: /advanced
        title: Advanced
```

App Structure - Comparison

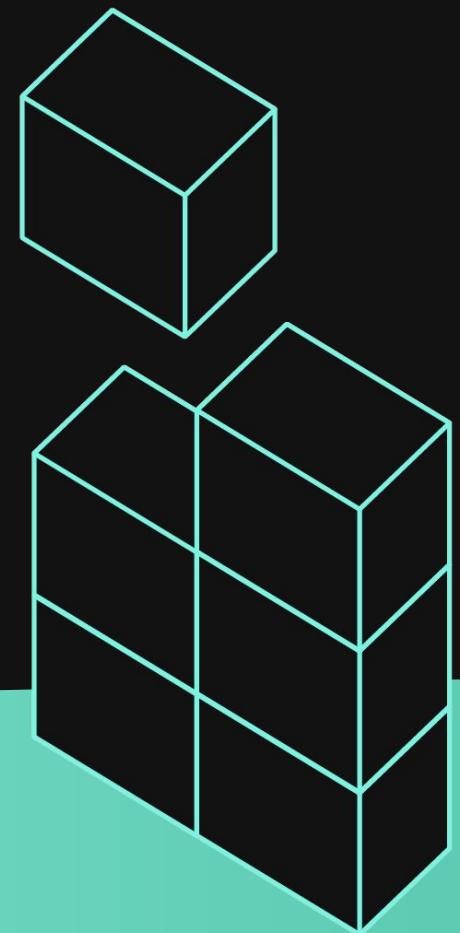
```
const routes = (
  <FlatRoutes>
    <Route path="/graphql" element={<GraphQLPage />} />

    <Route path="/settings" element={<UserSettingsPage />}>
      <SettingsLayout.Route path="/advanced" title="Advanced">
        <AdvancedSettings />
      </SettingsLayout.Route>
    </Route>
  </FlatRoutes>
)

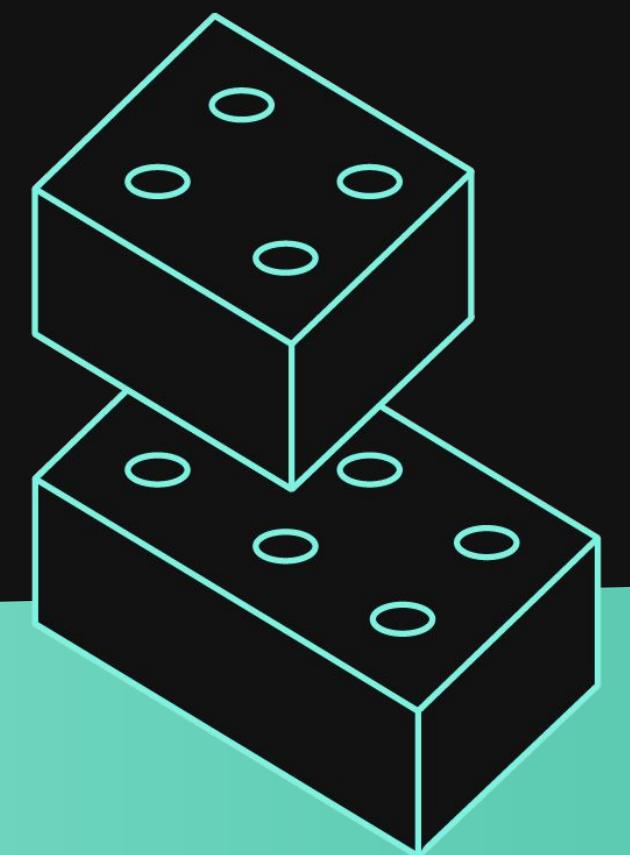
app.createRoot(
  <AppRouter>
    <Root>
      {routes}
    </Root>
  </AppRouter>
)
```



App Structure Demo



Usage

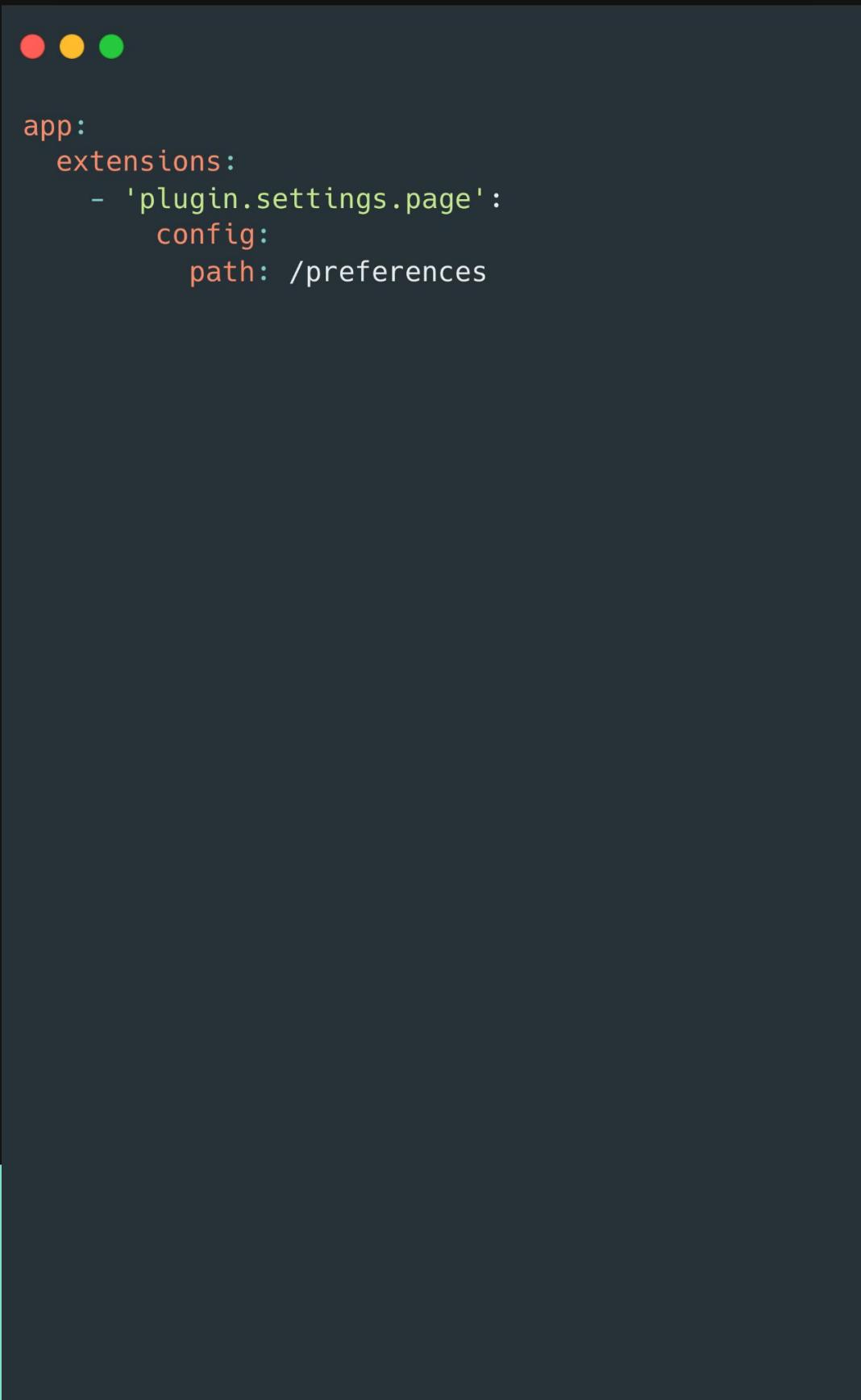


Configuration - Change Path

```
const routes = (
  <FlatRoutes>
    <Route path="/graphql" element={<GraphQLPage />} />

    <Route path="/preferences" element={<UserSettingsPage />}>
      <SettingsLayout.Route path="/advanced" title="Advanced">
        <AdvancedSettings />
      </SettingsLayout.Route>
    </Route>
  </FlatRoutes>
)

app.createRoot(
  <AppRouter>
    <Root>
      {routes}
    </Root>
  </AppRouter>
)
```



A screenshot of a Mac OS X application window titled "app". The window contains configuration code in JSON format:

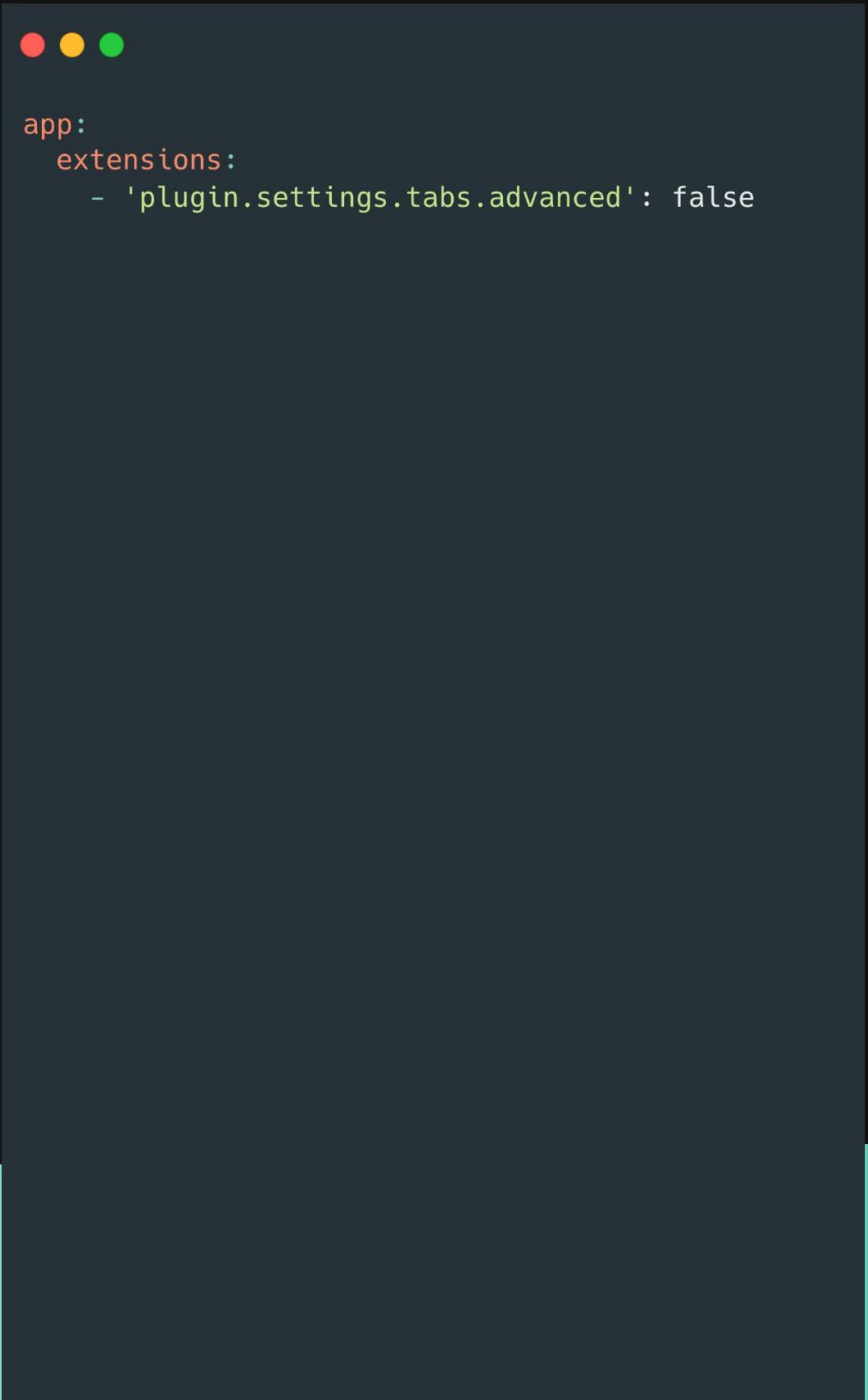
```
app:
  extensions:
    - 'plugin.settings.page':
      config:
        path: /preferences
```

Configuration - Disable

```
const routes = (
  <FlatRoutes>
    <Route path="/graphql" element={<GraphQLPage />} />

    <Route path="/preferences" element={<UserSettingsPage />}>
      {/* <SettingsLayout.Route path="/advanced" title="Advanced">
        <AdvancedSettings />
      </SettingsLayout.Route> */}
    </Route>
  </FlatRoutes>
)

app.createRoot(
  <AppRouter>
    <Root>
      {routes}
    </Root>
  </AppRouter>
)
```



A screenshot of a Mac OS X application window titled "app". The window contains configuration code for disabling the "Advanced" tab in the User Settings page. The code uses the `extensions` key in the `app` section to set the value of `plugin.settings.tabs.advanced` to `false`.

```
app:
  extensions:
    - 'plugin.settings.tabs.advanced': false
```

Creating an Extension



```
const examplePage = createPageExtension({
  id: 'plugin.example.page',
  defaultPath: '/example',
  loader: () => import('./ExamplePage').then((m) => <m.ExamplePage />),
})
```

Installing an Extension



```
const examplePage = createPageExtension({
  id: 'plugin.example.page',
  defaultPath: '/example',
  loader: () => import('./ExamplePage').then((m) => <m.ExamplePage />),
})

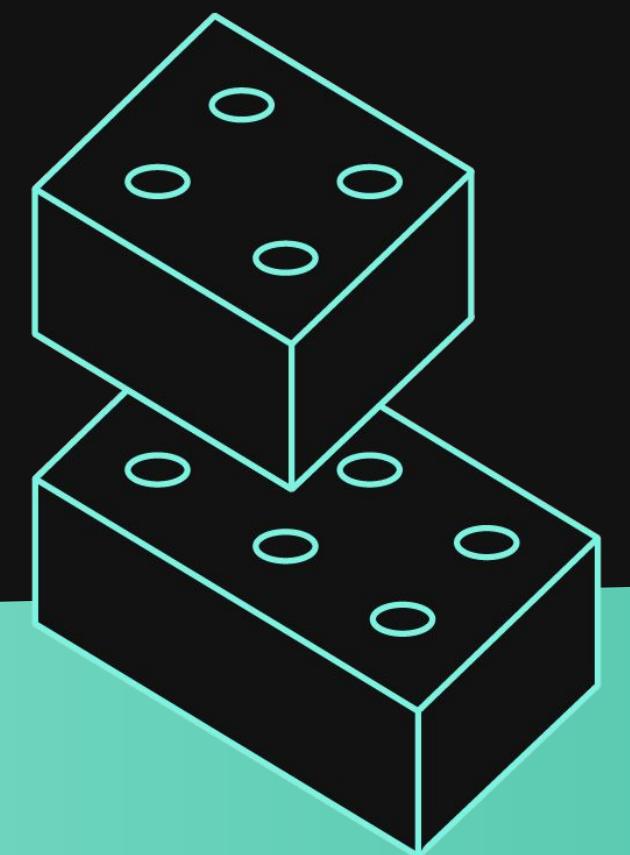
// @internal/plugin-example
export default createPlugin({
  id: 'example',
  extensions: [examplePage],
})
```

Creating an Extension

Low Level API

```
export const coreLayout = createExtension({
  id: 'app.layout',
  attachTo: { id: 'app', input: 'root' },
  inputs: {
    nav: createExtensionInput(
      { element: coreExtensionData.reactElement },
      { singleton: true },
    ),
    content: createExtensionInput(
      { element: coreExtensionData.reactElement },
      { singleton: true },
    ),
  },
  output: {
    element: coreExtensionData.reactElement,
  },
  factory: ({ inputs }) => ({
    element: (
      <SidebarPage>
        {inputs.nav.element}
        {inputs.content.element}
      </SidebarPage>
    )
  }),
})
```

Extension Demo



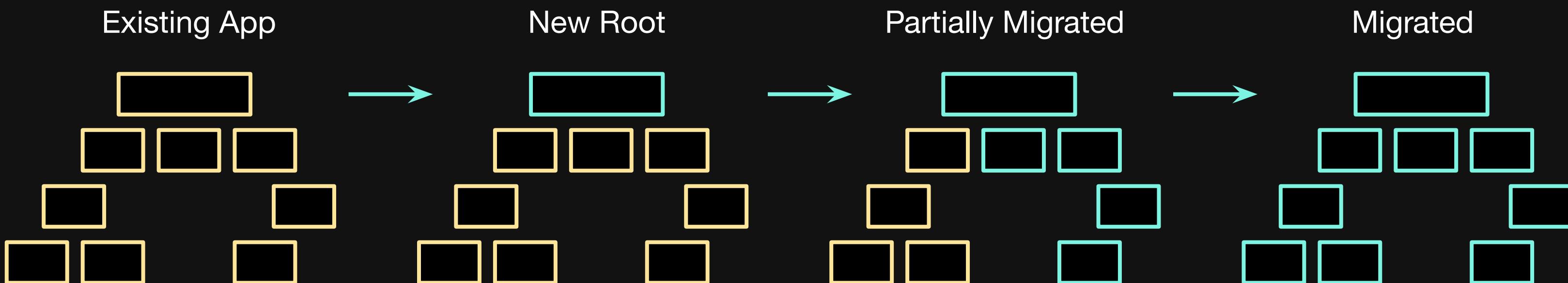
Compatibility



Plugin Compatibility



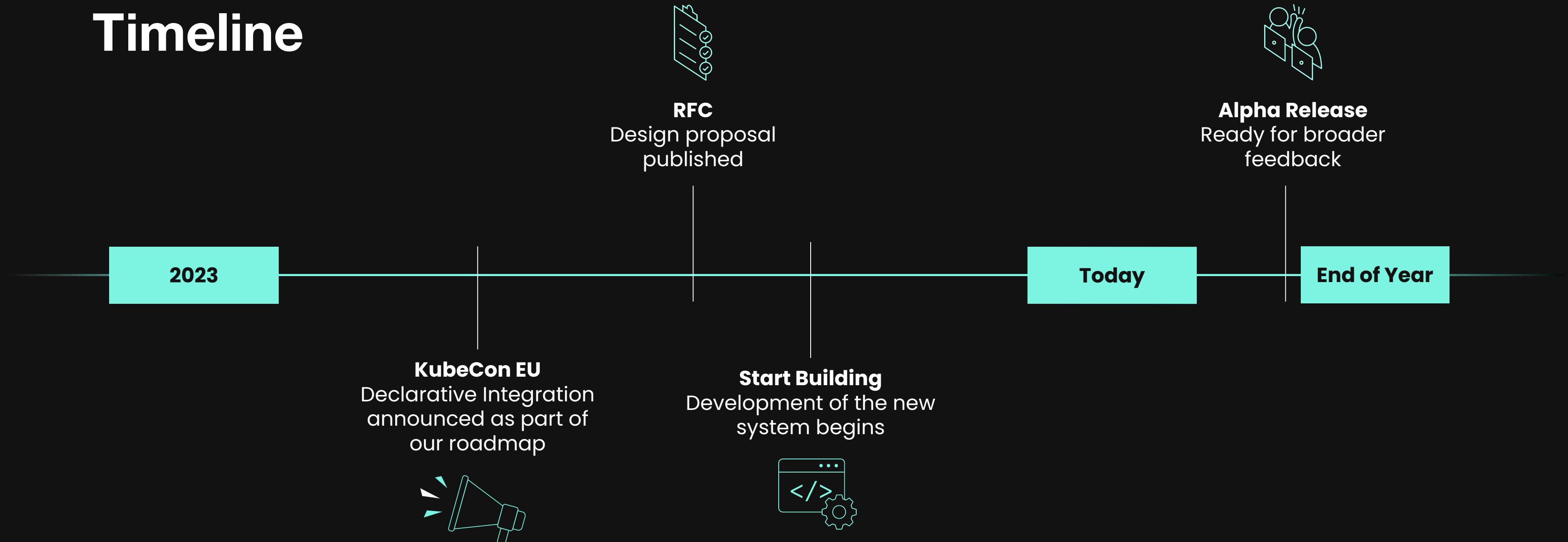
App Migration



Compatibility Demo



Timeline





Thank you!

Docs, demos, and updates

backstage.io

Adopting Backstage?

backstage.spotify.com

frontside.com/backstage

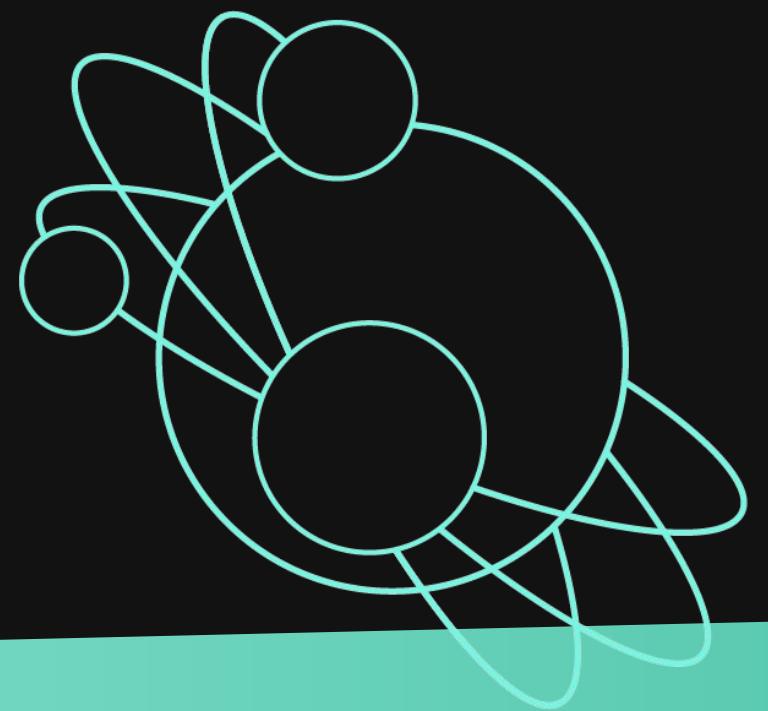
roadie.io

thoughtworks.com

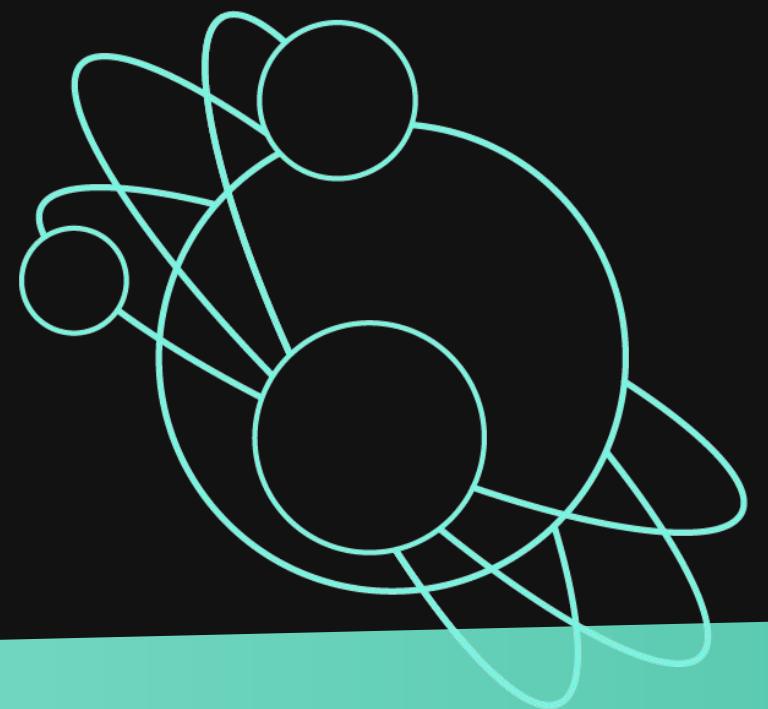
Feedback



OUTRO



SKIPPED



Lines of TypeScript

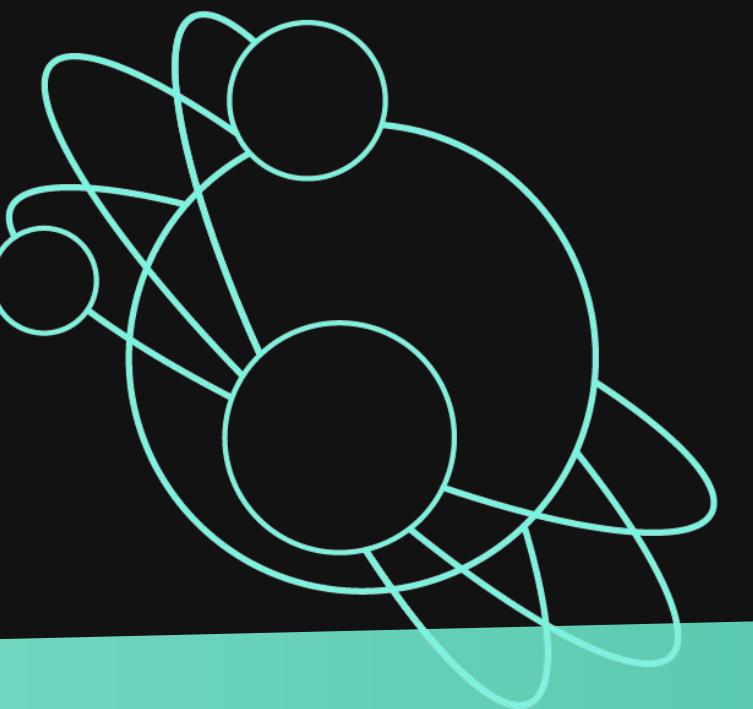




i18n

Translations & Message Customization

Contributed by [@mario-mui](#)



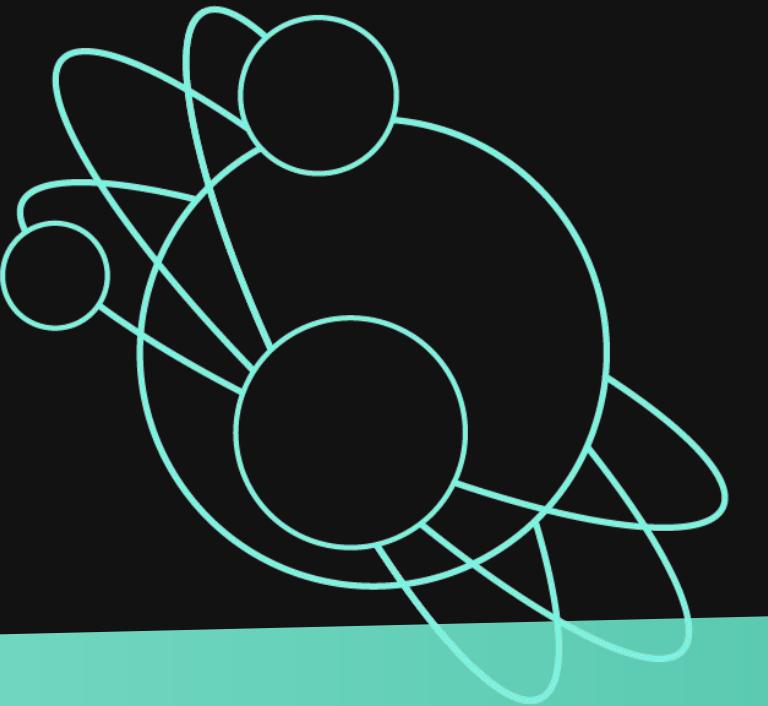
Message Overrides



```
createTranslationMessages({  
  ref: catalogTranslationRef,  
  messages: {  
    catalog_page_create_button_title: 'Register',  
  }  
})
```



OpenAPI Tooling





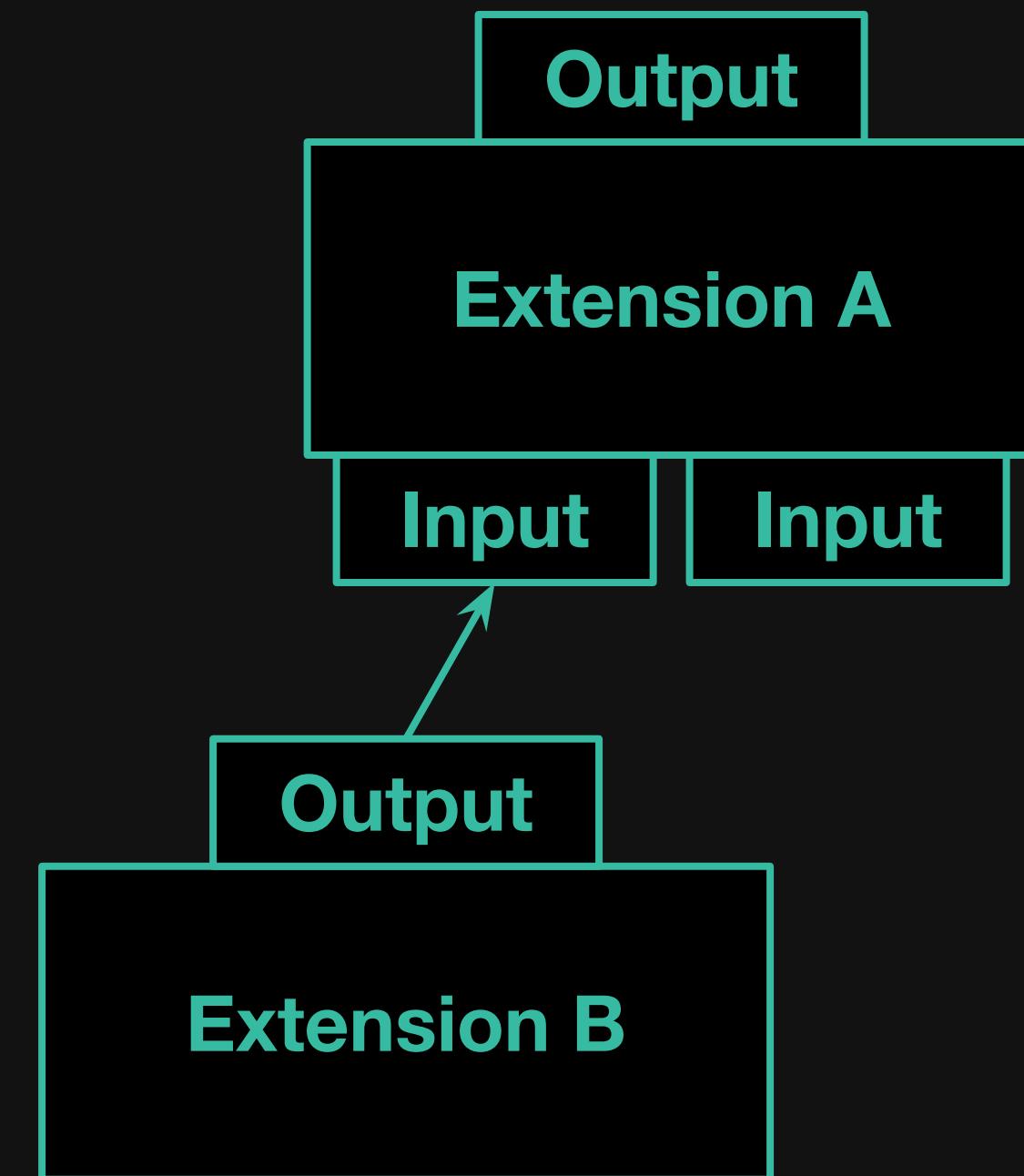
Design Priority

Integrator -> Plugin Builder -> Maintainer

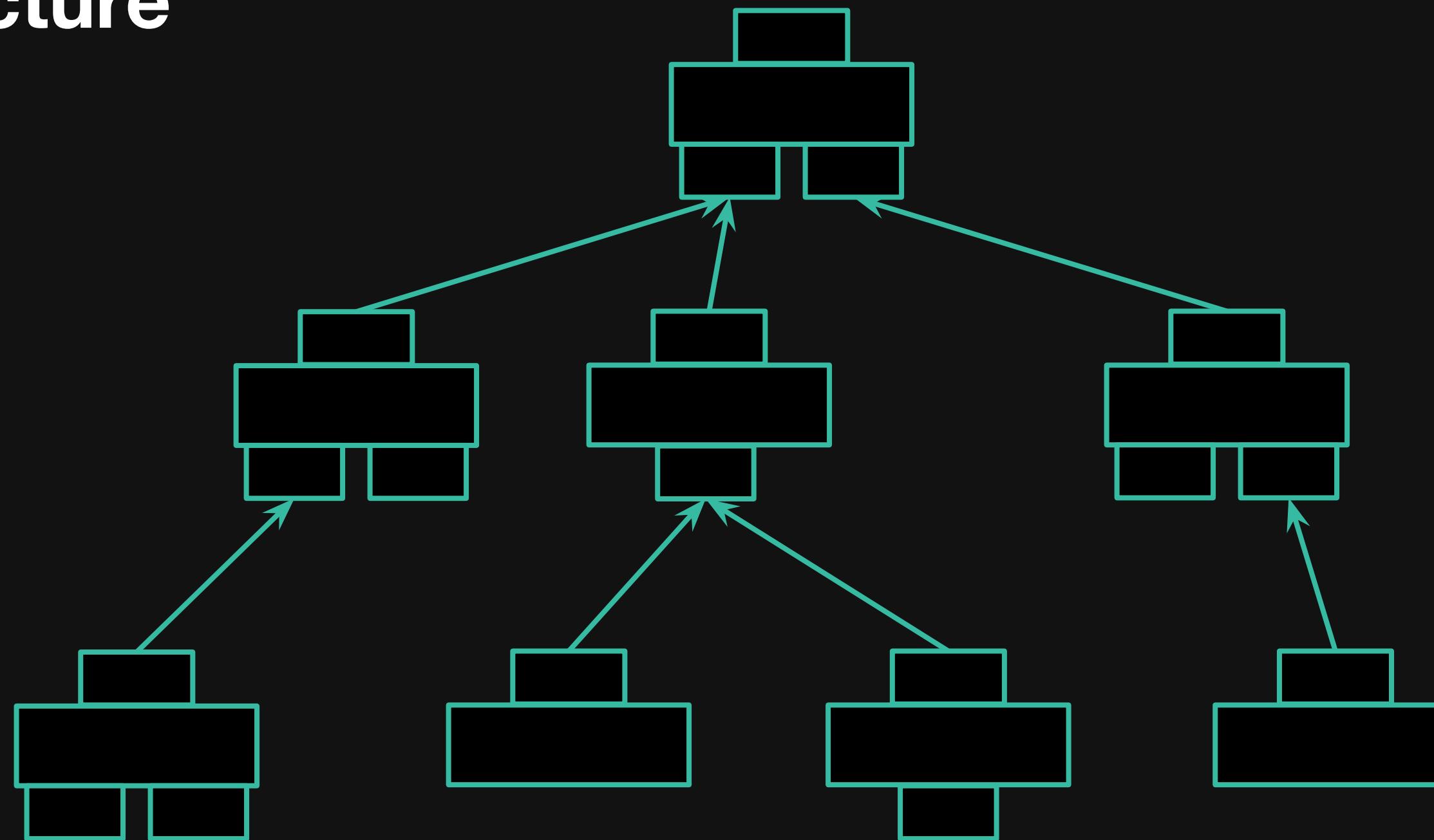
Communication



Communication



App Structure





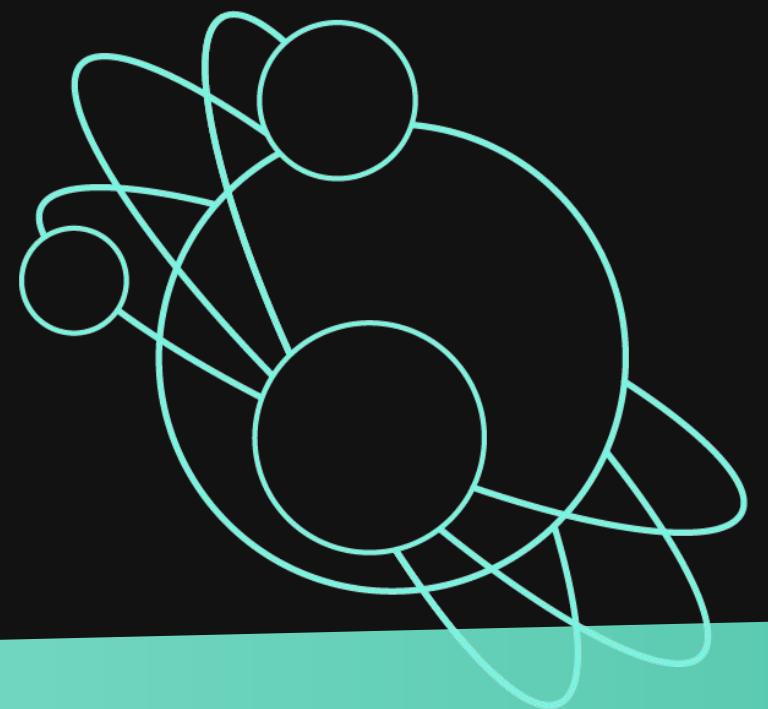
Communication



```
<Route path="/settings" element={<UserSettingsPage />}>
  <SettingsLayout.Route path="/notifications" title="Notifications">
    <NotificationSettings />
  </SettingsLayout.Route>

  <SettingsLayout.Route path="/advanced" title="Advanced">
    <AdvancedSettings />
  </SettingsLayout.Route>
</Route>
```

TEMPLATES



Code Slide Template



```
const backend = createBackend()

backend.add(scaffolderPlugin());
backend.add(scaffolderModuleGitlabActions());
backend.add(scaffolderModuleCookiecutterActions());

await backend.start()
```

Figure Slide Template

