

# React Hooks

Oliver Sturm • @olivers[@fosstodon.org] • oliver@oliversturm.com



# Oliver Sturm

- Training Director at DevExpress
- Consultant, trainer, author, software architect and developer for 30 years
- Contact: [oliver@oliversturm.com](mailto:oliver@oliversturm.com)

# Agenda

- Hooks vs HOCs - what's the point?
  - (Note the amazing wordplay in the previous bullet)
- Standard hooks
- Custom hooks
- Hooks in 3rd party libraries

# What's the point of React Hooks?

- React has a functional structure with a side-effect free rendering mechanism
- Preference for functional component implementation
  - Class-based components still remain available!
- HOCs always supported for modularization
- Complex class and HOC structures end up with deep nesting
- Hooks: functional state management based on closures
- Popular by now in other JS-based environments

# The Hook Rules

```
const MyComponent = () => {  
  // Only use hook calls at the top level (that's rule 1)  
  // of React functions (that's rule 2).  
  // Note that "React functions" include custom hooks.  
  
  const [val, setVal] = useState();  
  
  return <div>Val is {val}</div>;  
};
```

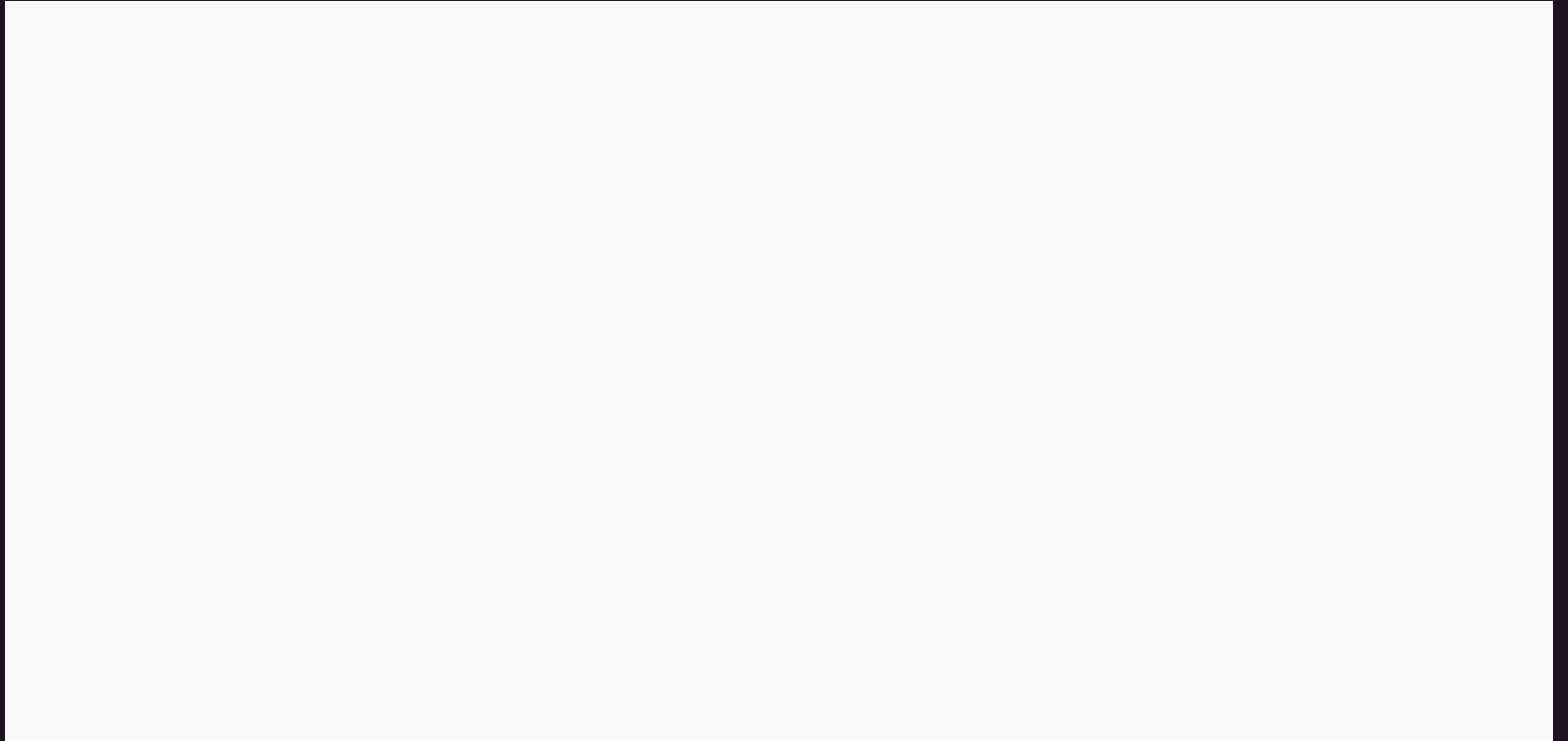
# Fragen

- Cloud - Kostenkontrolle - wie weiss man, was die eigene Anwendung kostet?
- Hochverfuegbarkeit notwendig - kann man "hybrid" oder lokal arbeiten, wenn die Cloud unzugänglich ist? Docker - Registry lokal?
- On Premise-System in die Cloud migrieren, moeglichst ohne Aenderungen, was sind die Stolpersteine?
- GraphQL - wo gehoert das ins Datenzugriffssystem? Im Vergleich mit

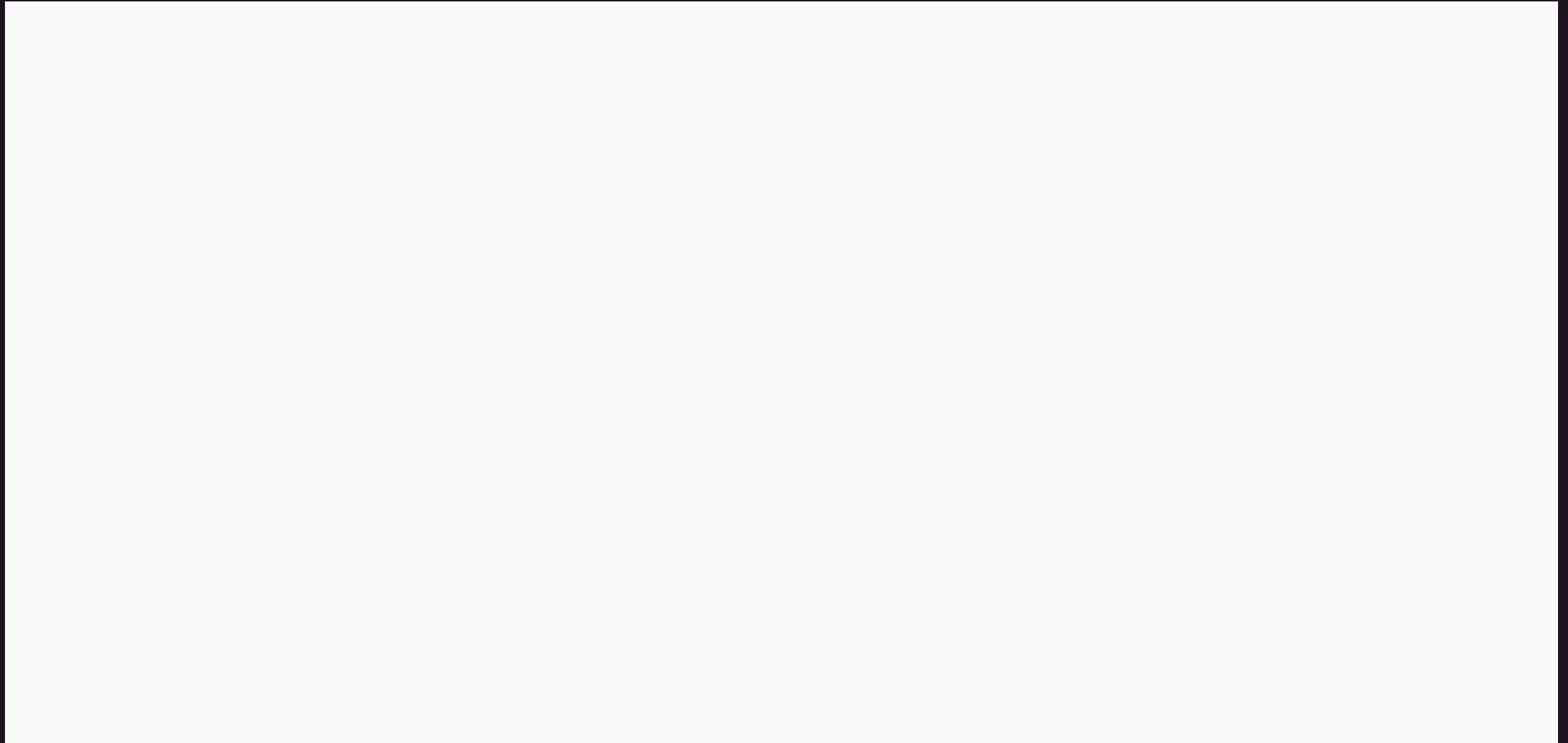
OData?

- Daten transferieren zwischen lokal und Cloud im Hybridsetup? Datenbankinhalte
- Beispiel Consul - Azure-Service oder separater Container - SaaS oder separates Deployment?

# Vor COVID - Eingeschränkter Remotezugriff

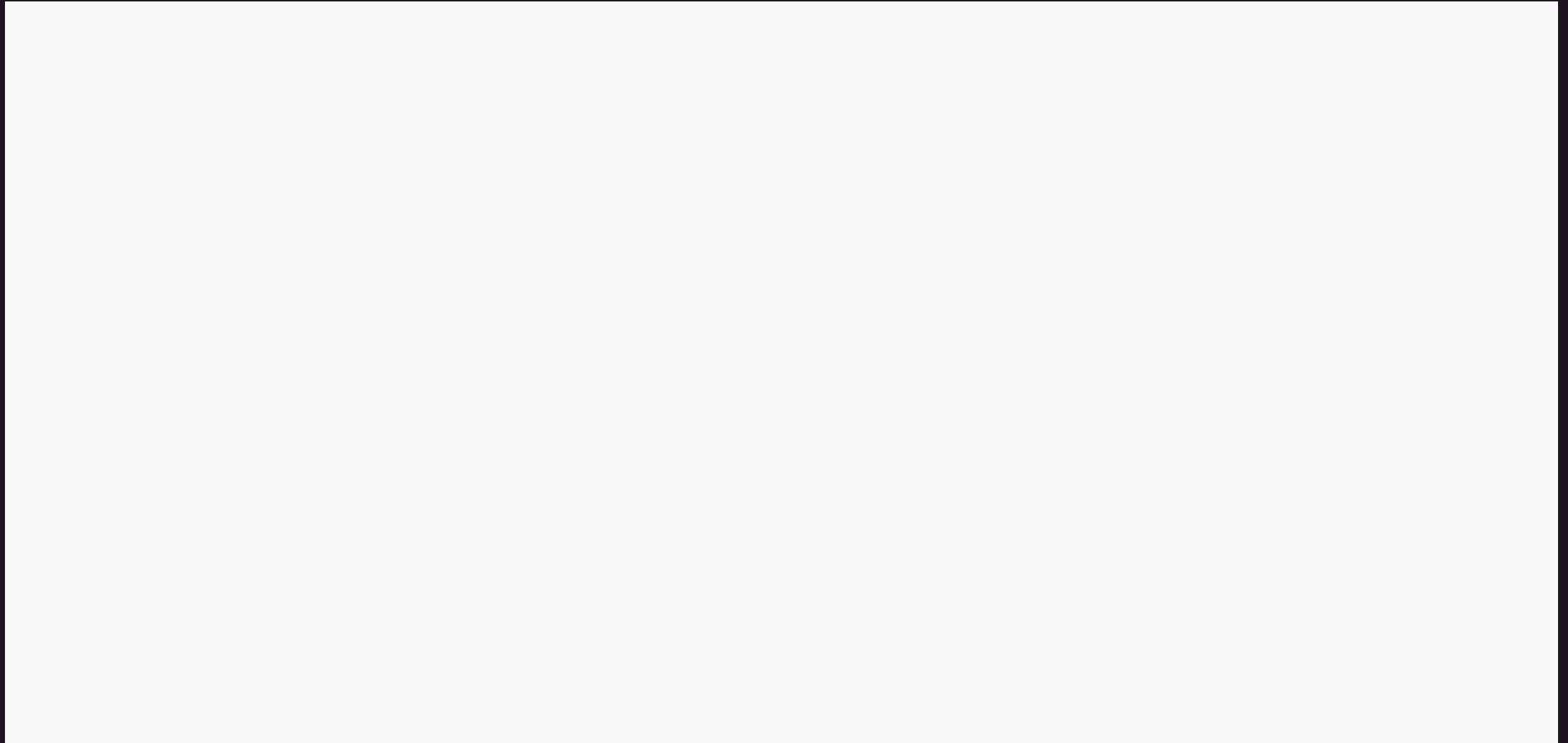


# Vor COVID - Eingeschränkte Remote-Funktionalität

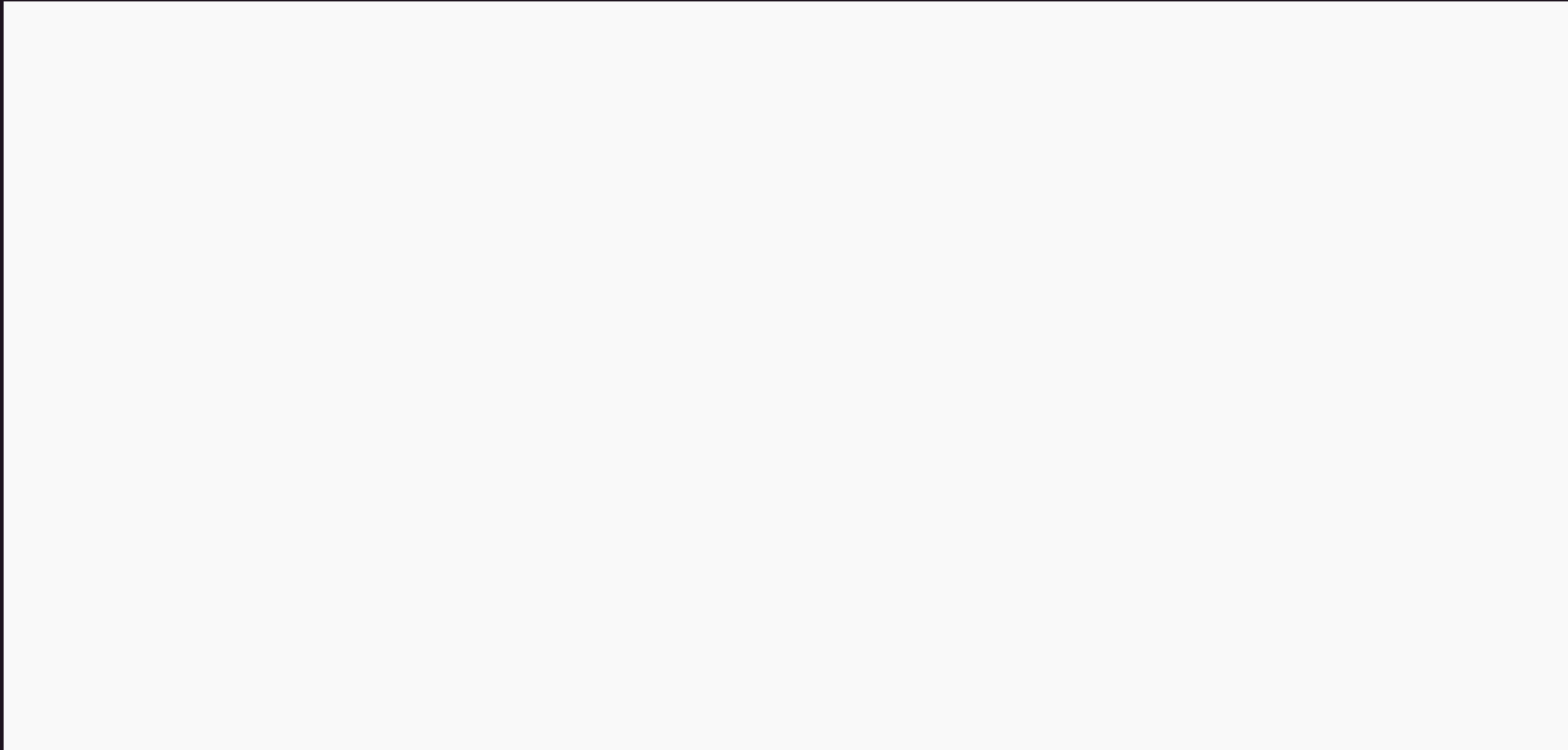




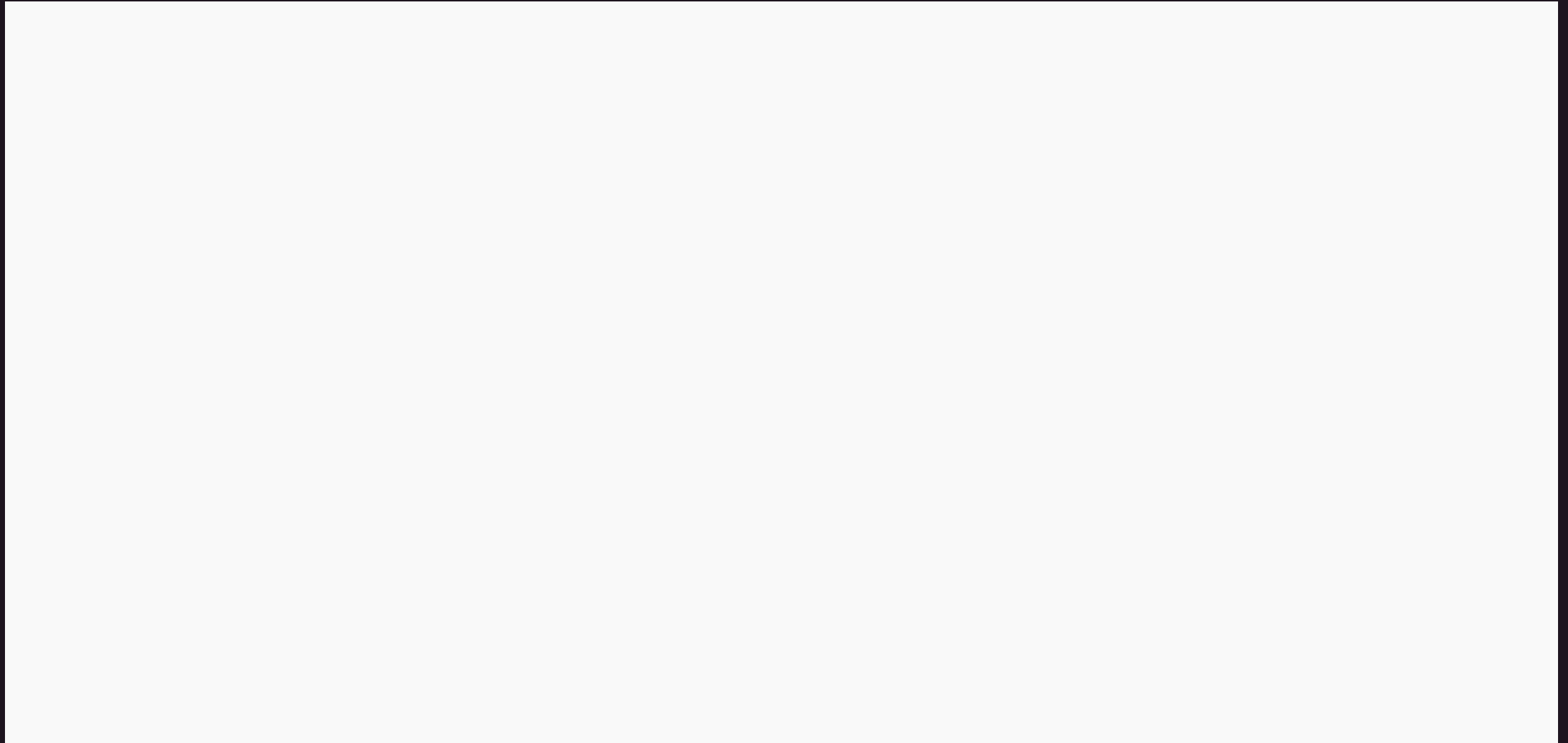
# Fernzugriff auf eine Datenbank



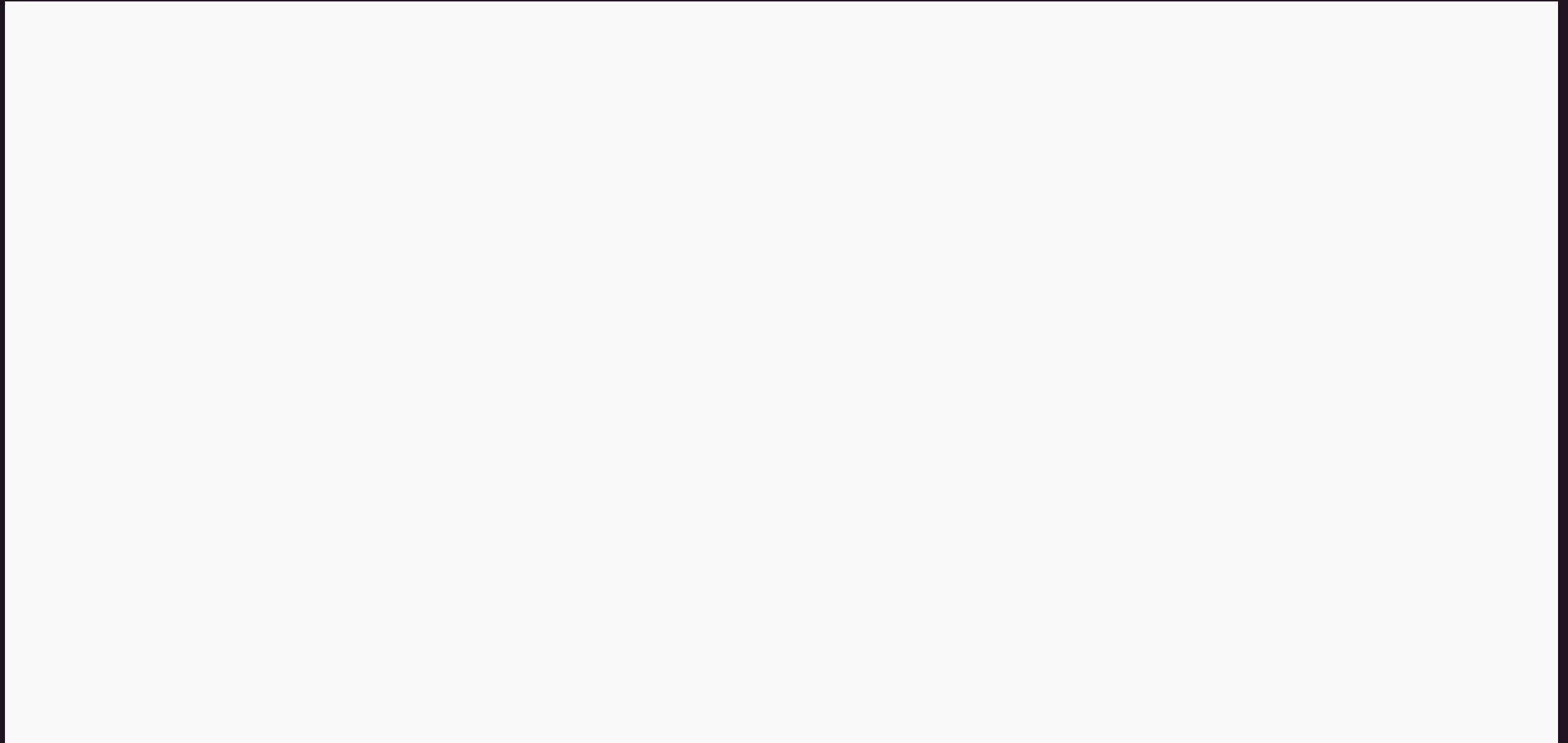
# Fernzugriff mit Remote Desktop



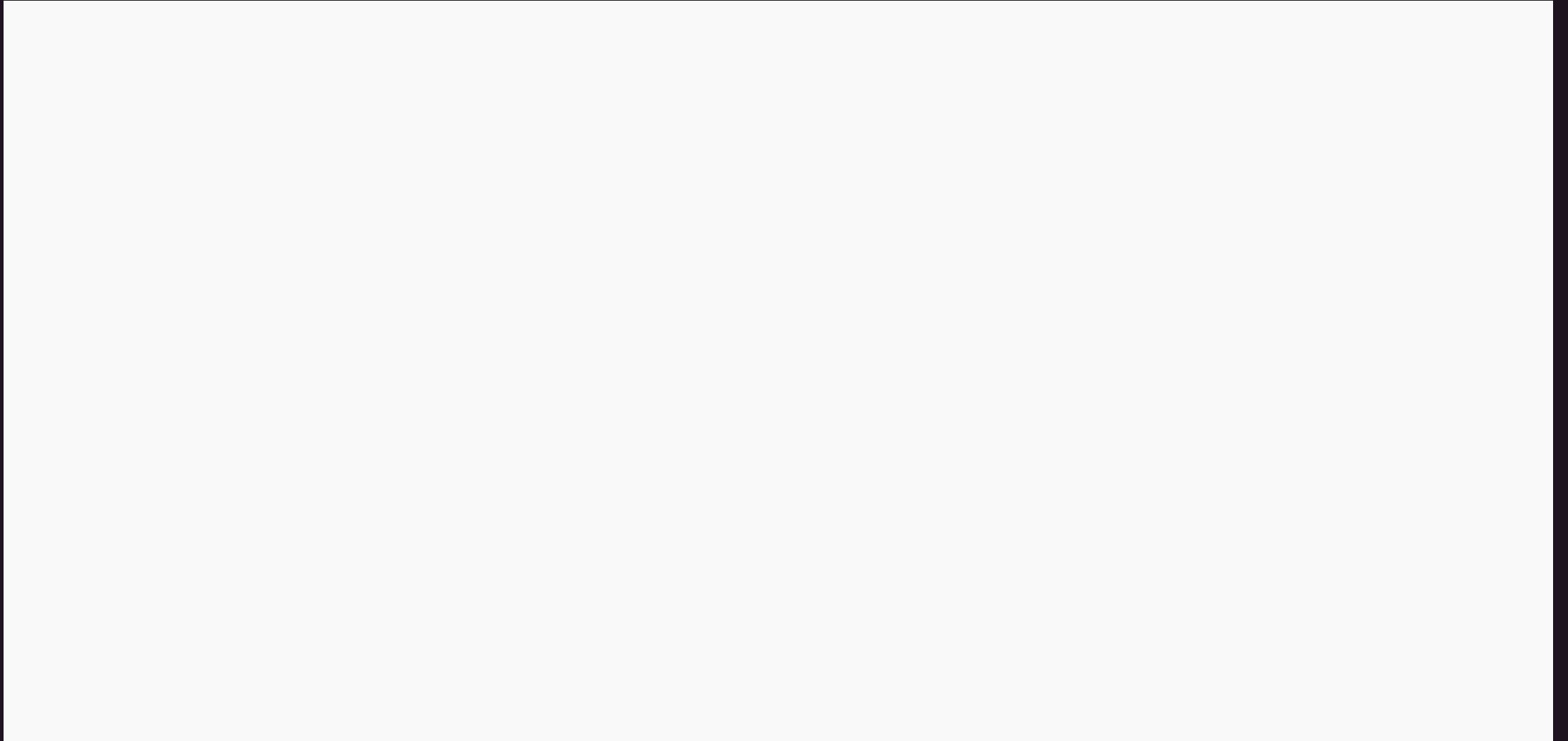
# Einfache Web-Anwendung

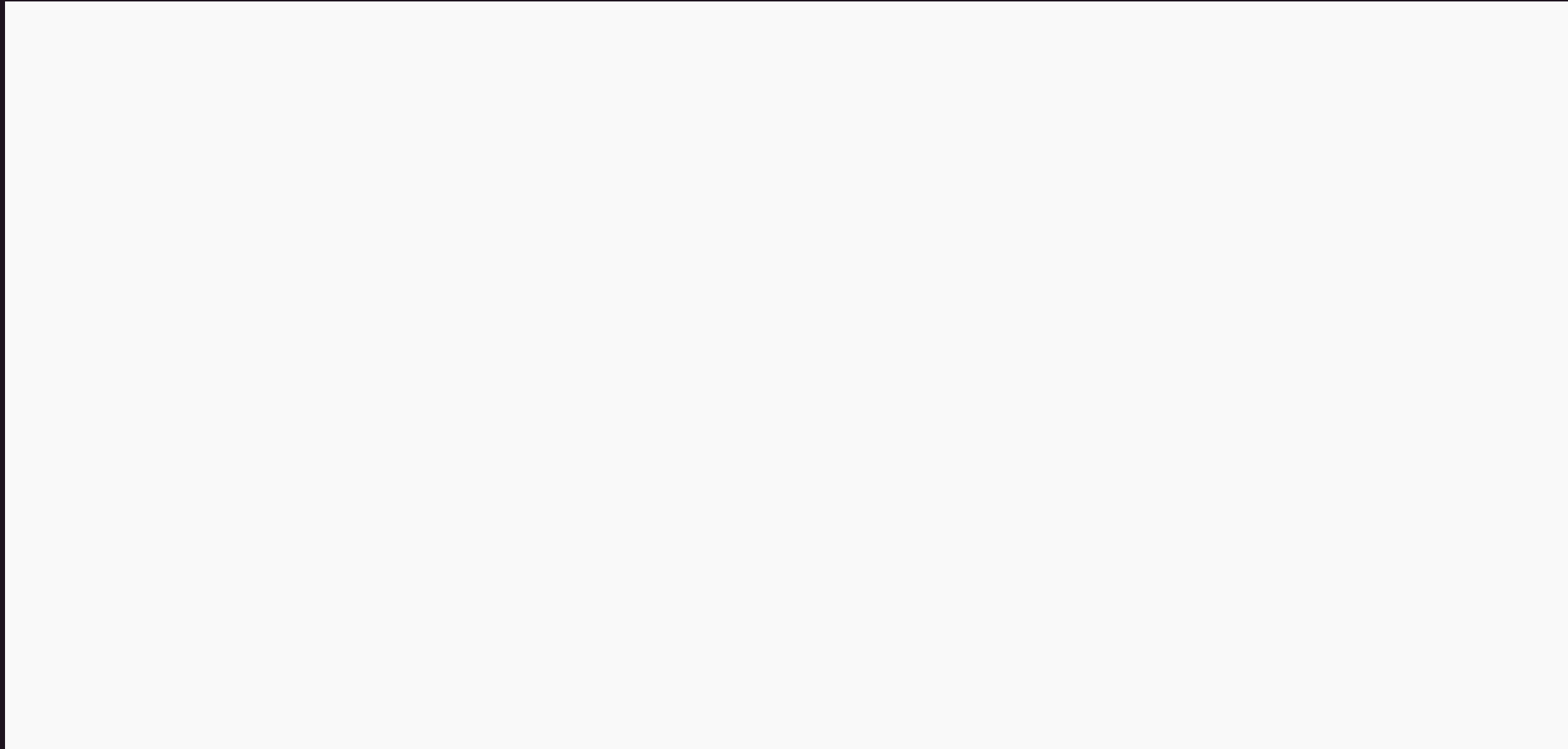


# (Micro-?) Services

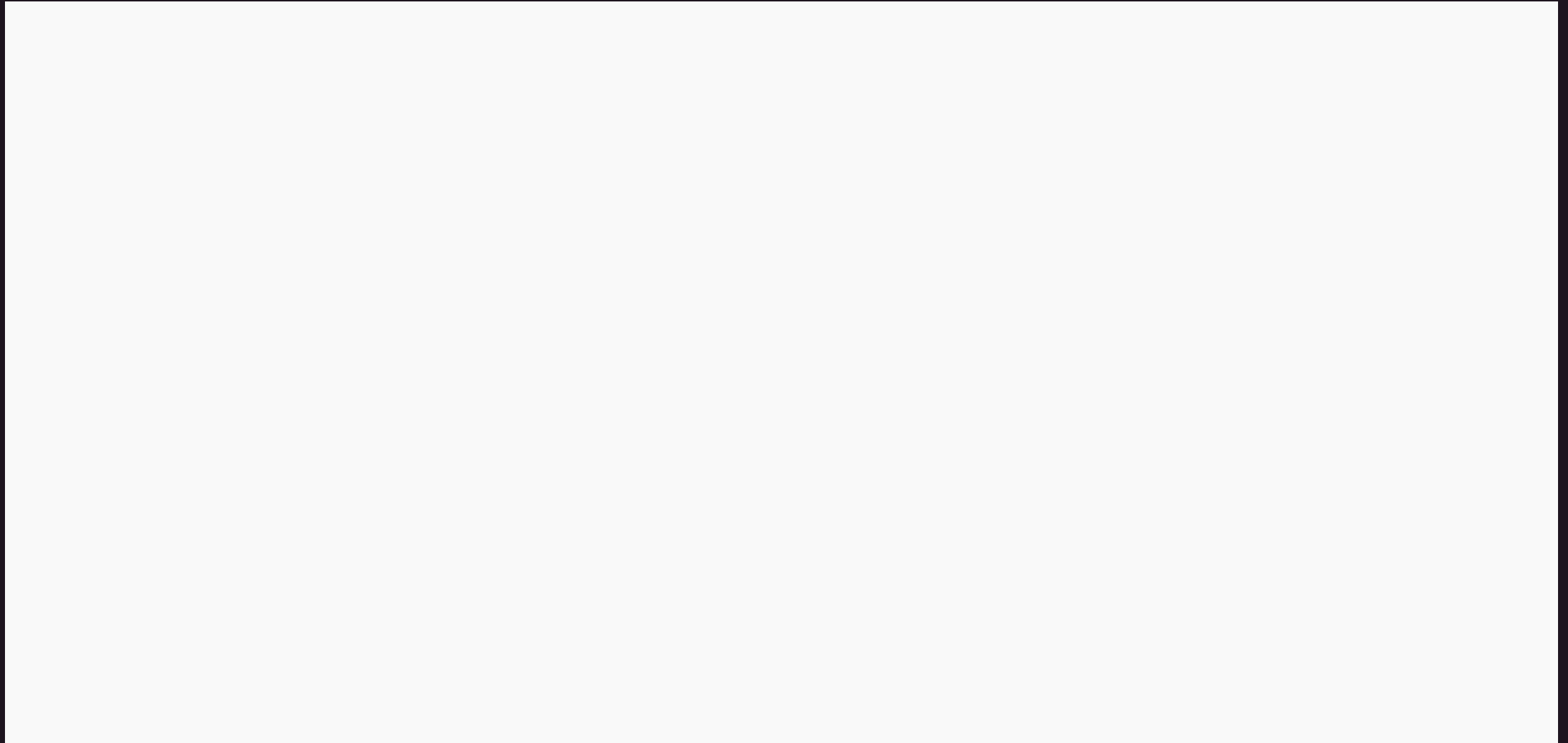


# Dienststruktur - Micro oder Größer

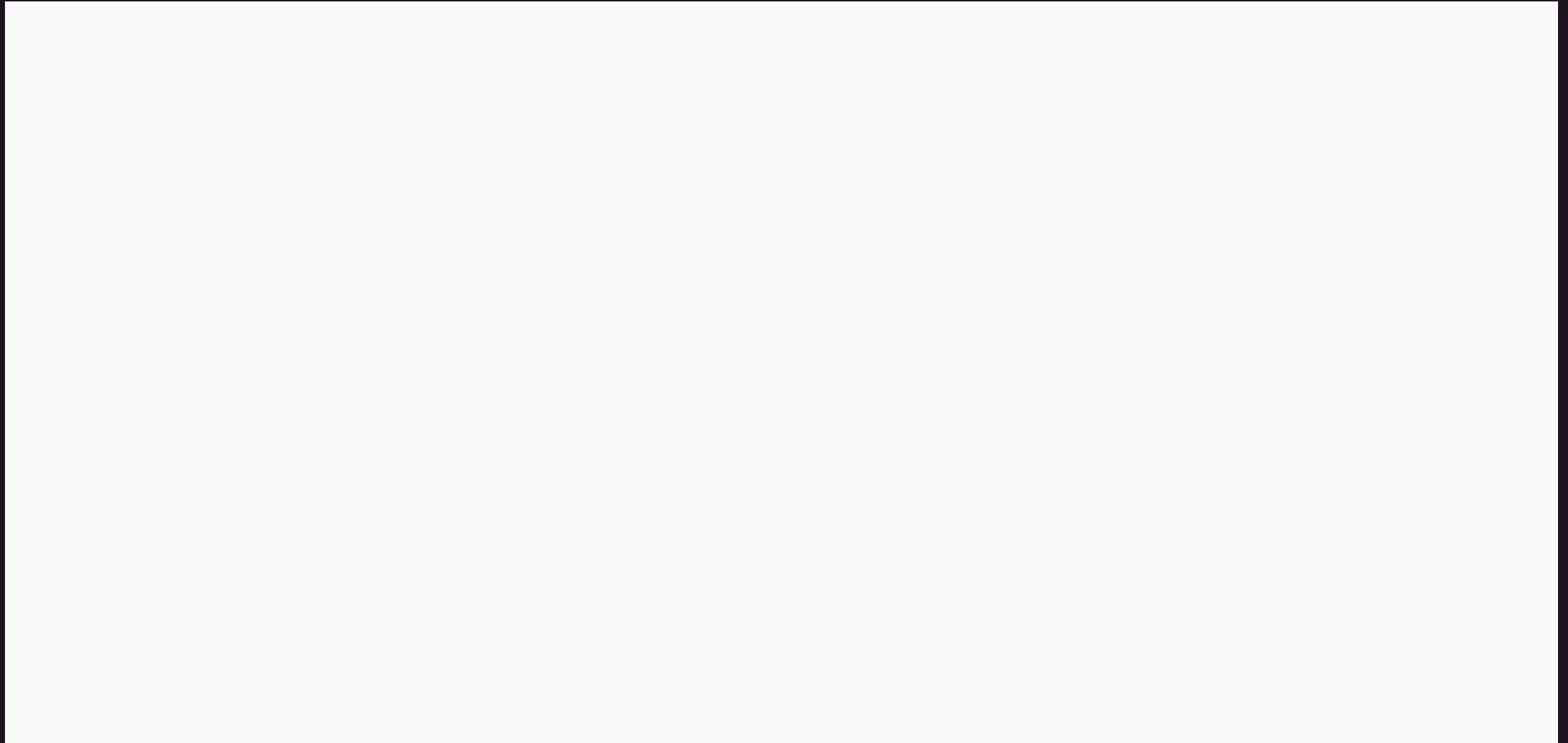




# CQRS mit Event Sourcing

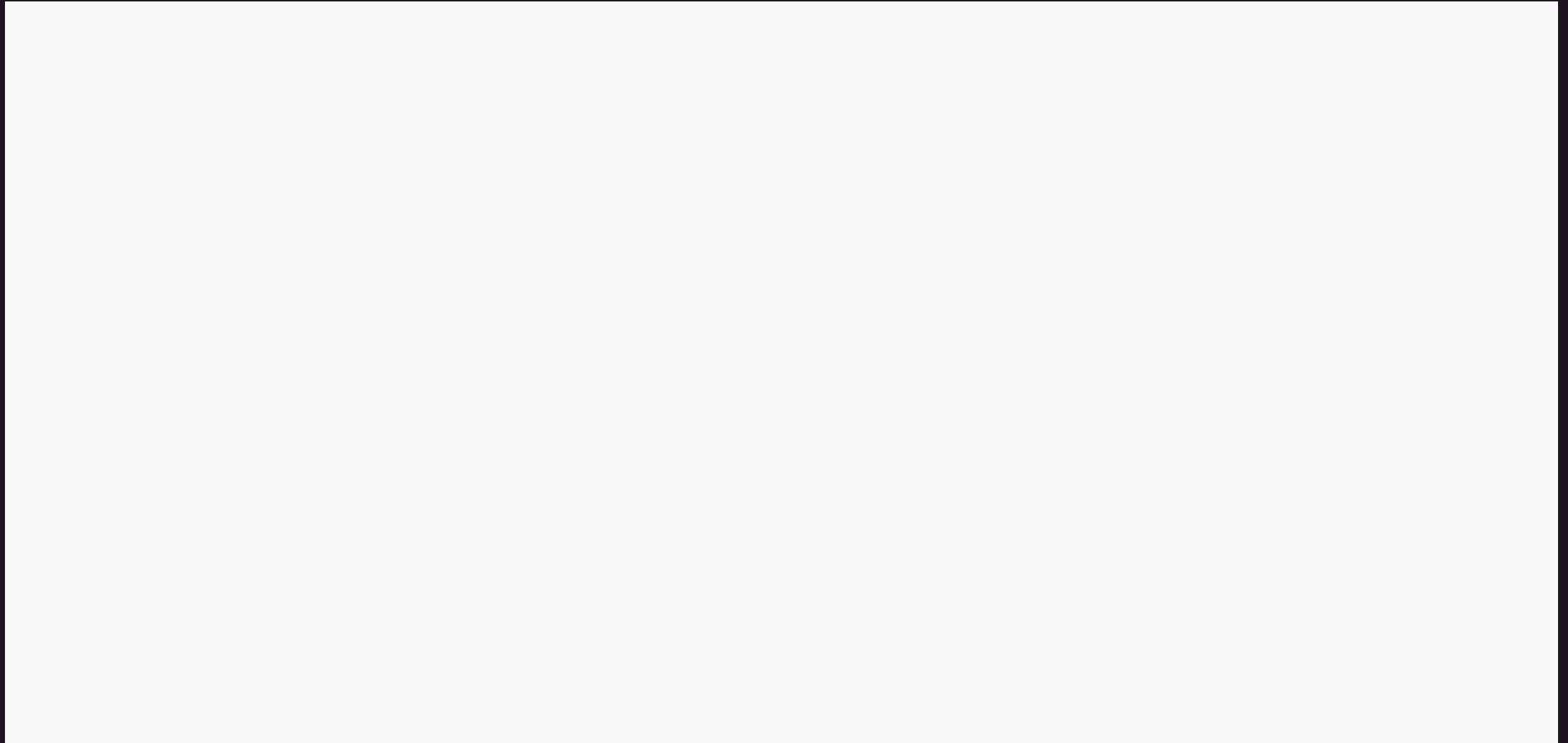


# GraphQL Basics

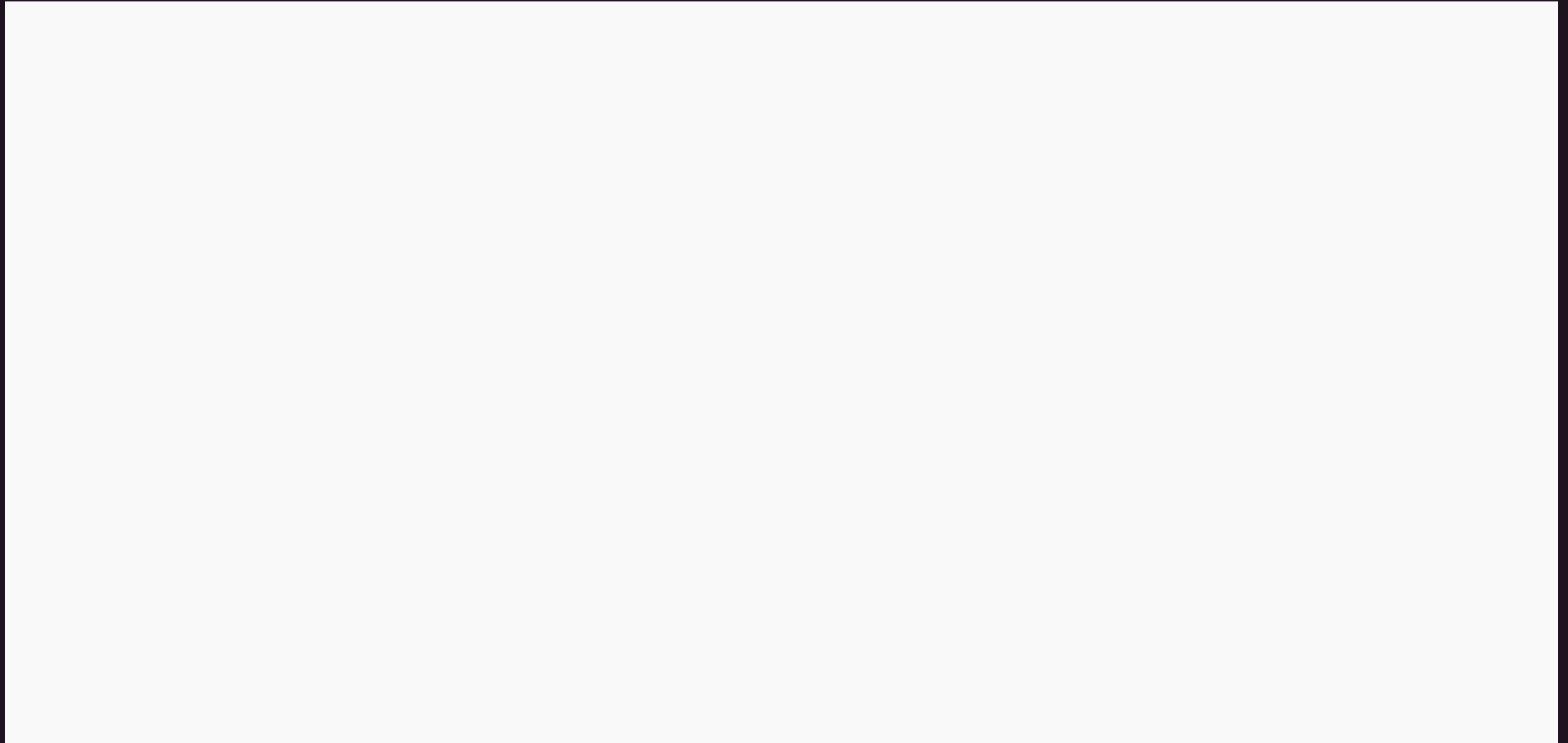




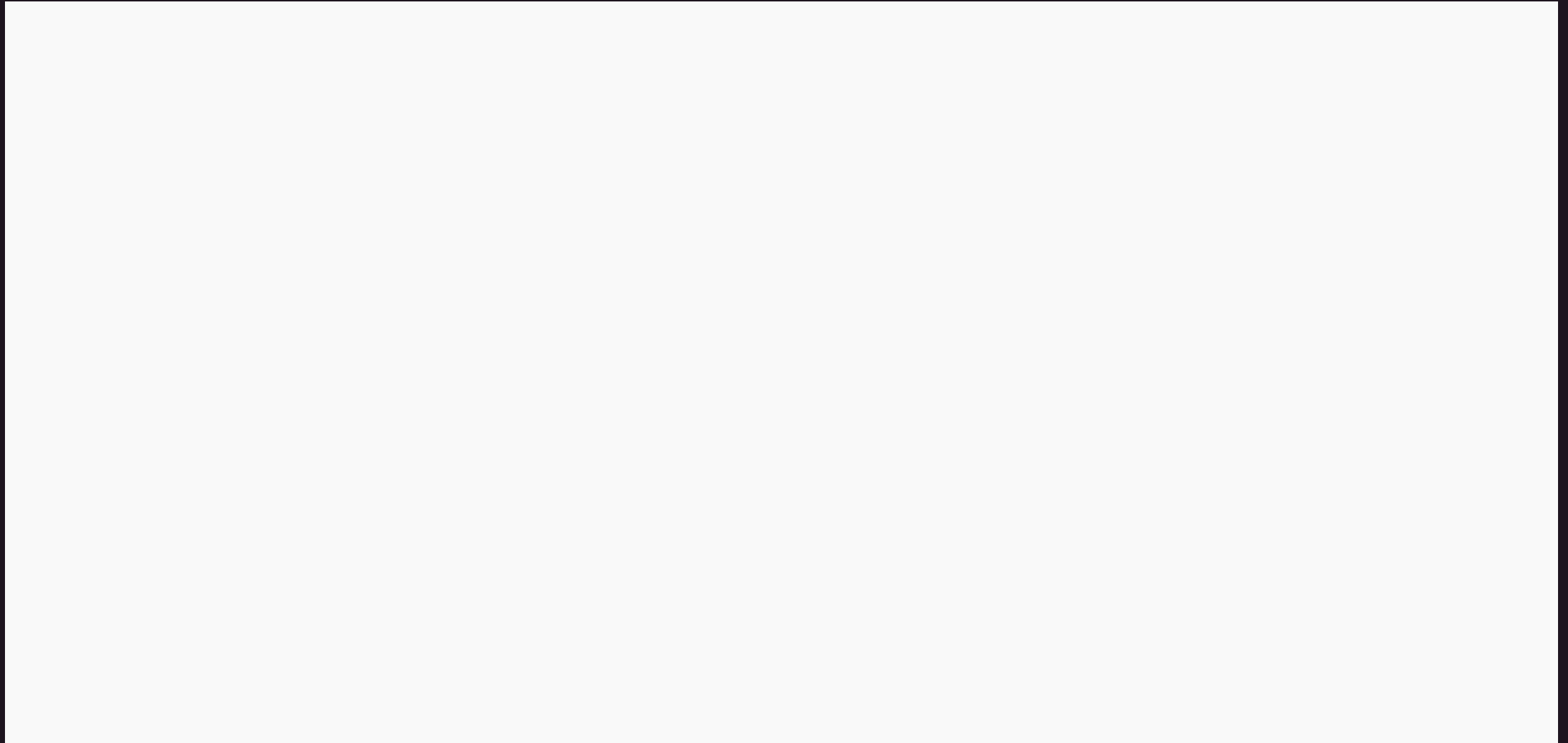
# GraphQL Endpoint in Readmodel



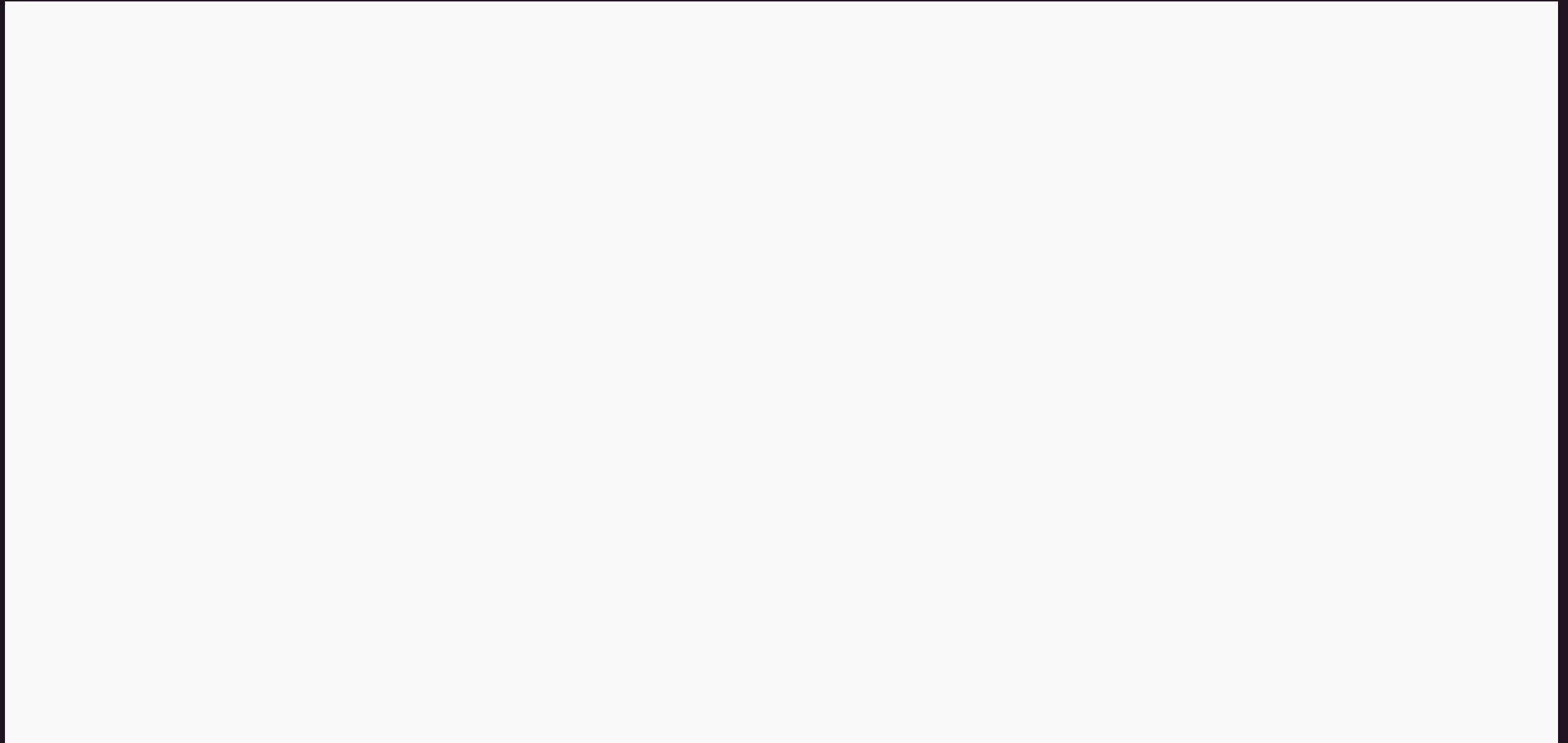
# GraphQL - Frontend Service



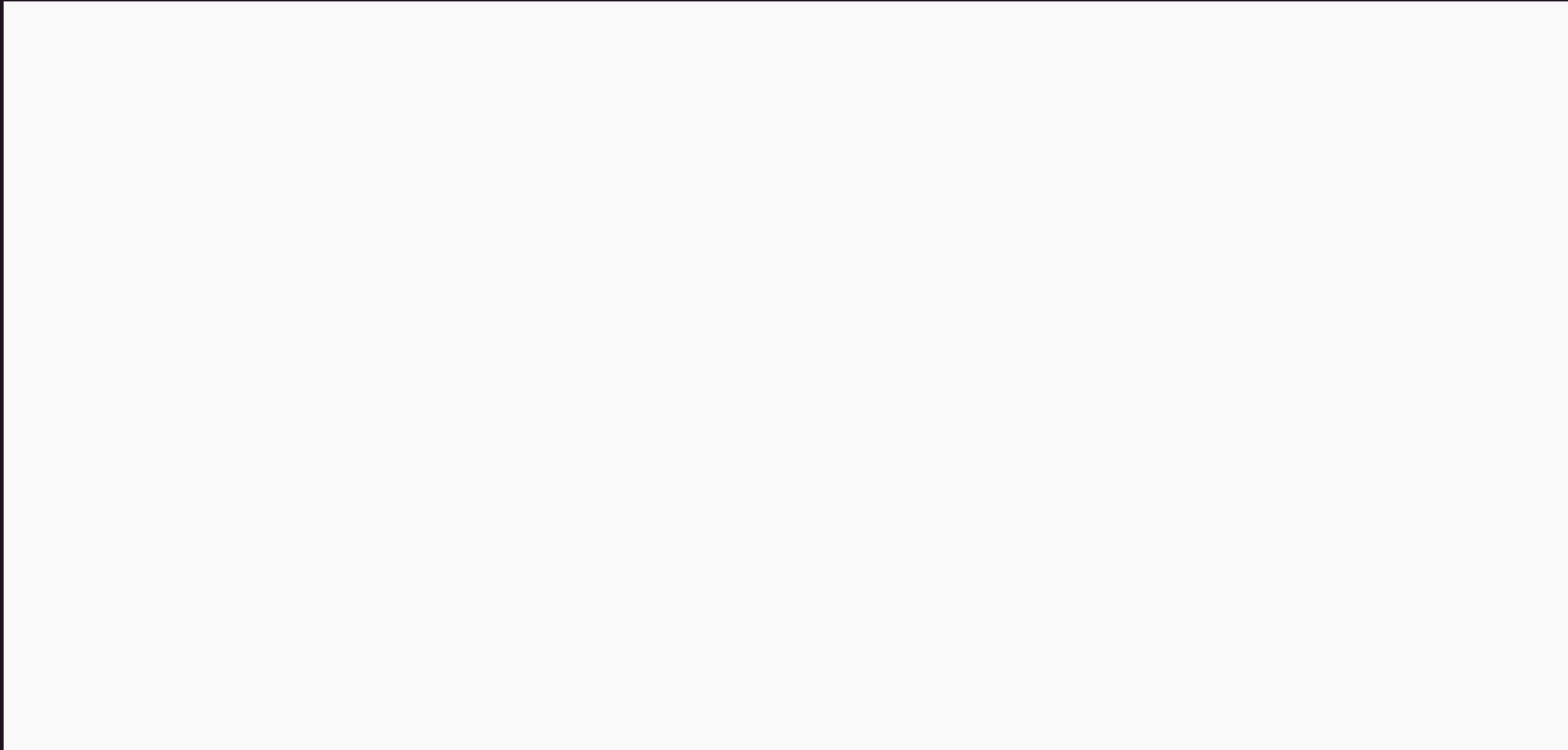
# Hypervisors



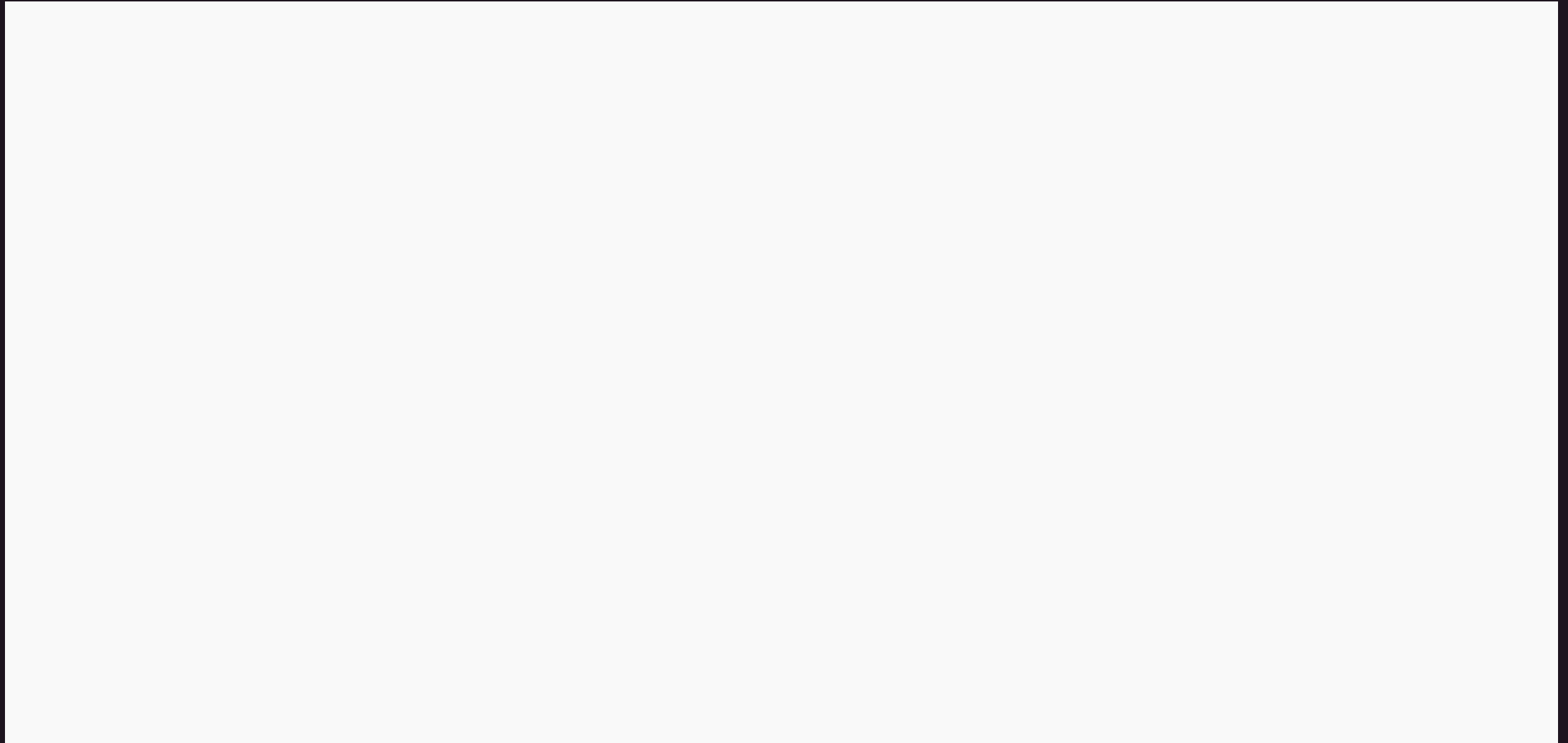
# Container und VMs



# CAP Theorem



# JavaScript vs TypeScript



# Sources

- This presentation:
  - <https://oliversturm.github.io/developers-and-architects/basta-2020>
  - PDF download:  
<https://oliversturm.github.io/developers-and-architects/basta-2020/slides.pdf>

# Thank You

Please feel free to contact me about the content anytime.

Oliver Sturm • @olivers[@fosstodon.org] • oliver@oliversturm.com

