

NEXT.JS

@



digitec.ch



GALAXUS

About Us

Christian Sany

Senior Frontend Engineer

Editorial Content

Shop Special Deals

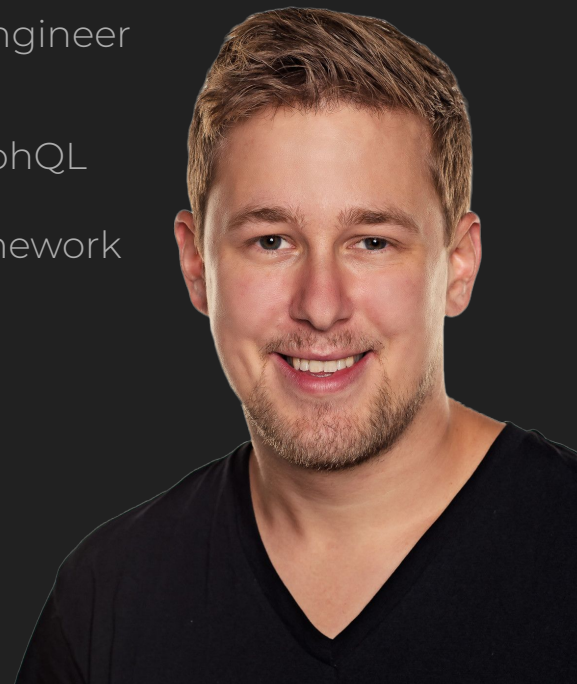


Christoph Ebert

Lead Software Engineer

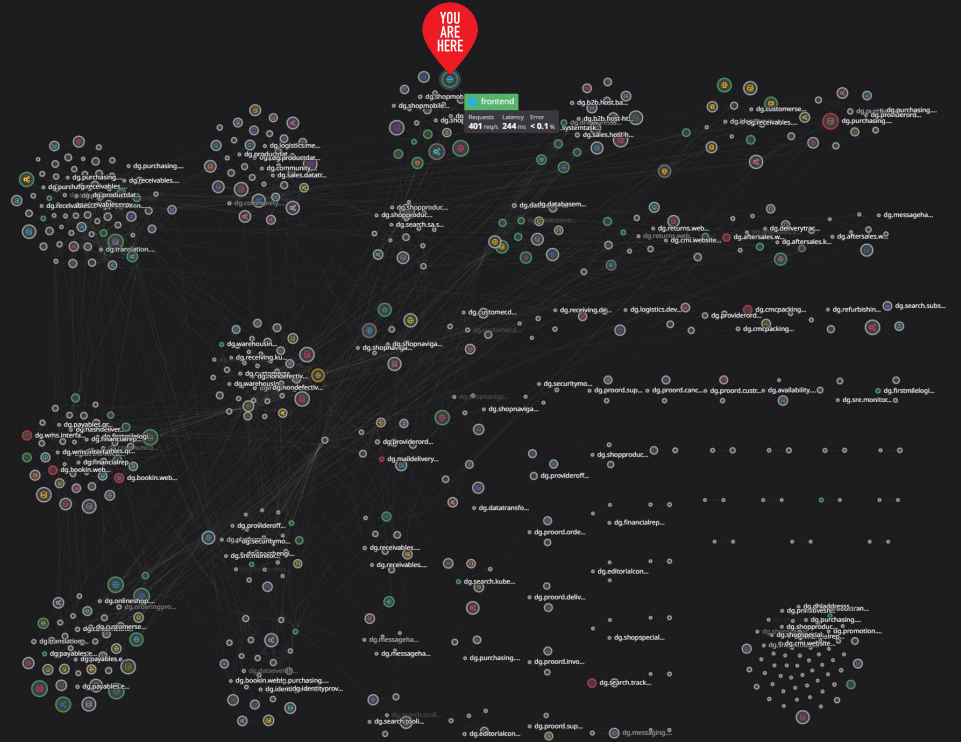
Frontend- & GraphQL

Application Framework



Digitec Galaxus

- 2700+ Employees
- ~300 Devs in Product Development
- 16 Online Shop Scrum Teams
- 30 Frontend Devs



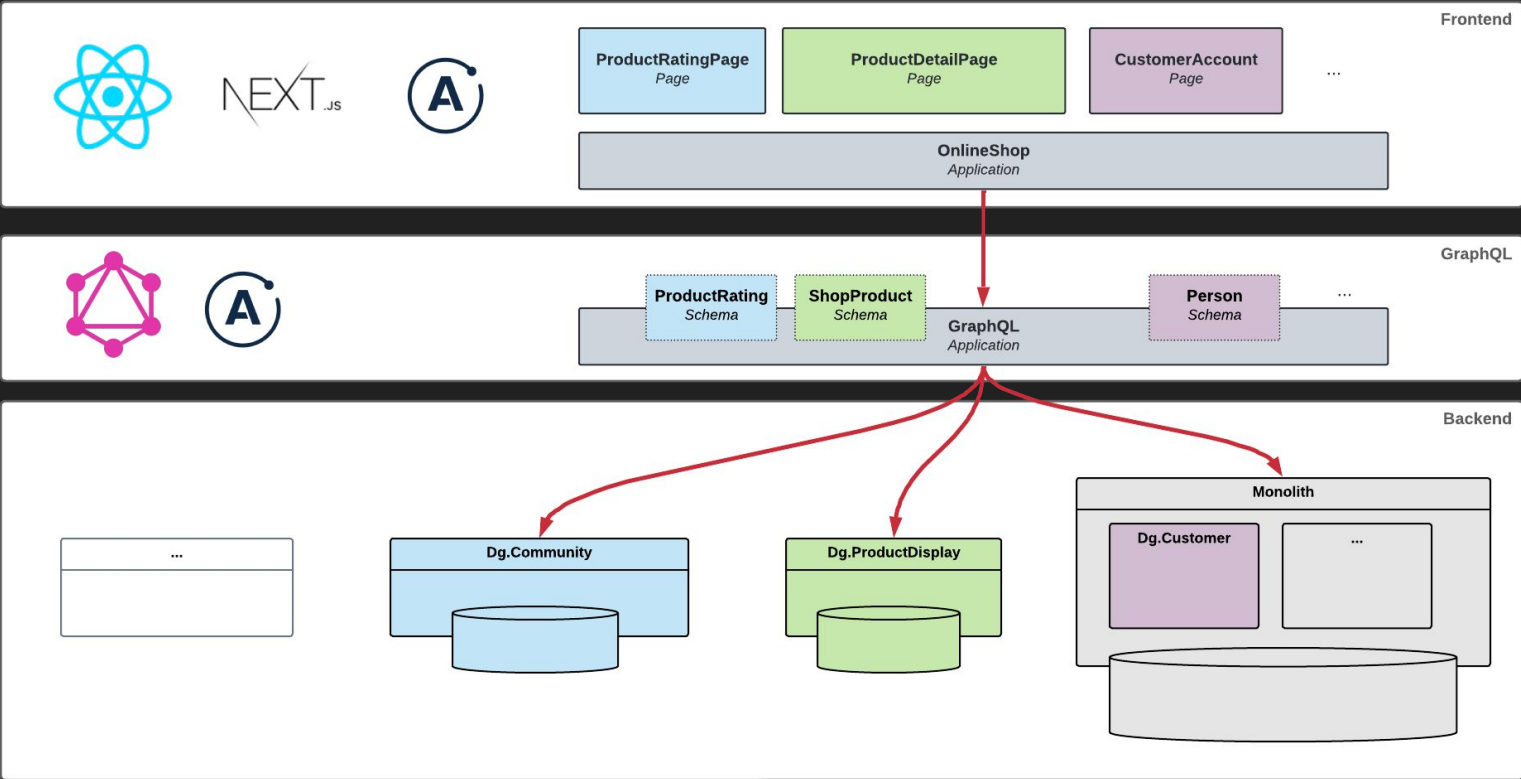
Why Next.js?

Built-in 
on top of React

Collaboration at scale

Flexible Rendering Strategies

Shop Architecture



Frontend Workflow

- Automation: Continuous deployments
- Fail fast: Pull-Request Checks
- Quick feedback: Preview Environment

The screenshot shows a pull request interface with two main sections. The top section, titled 'Christoph Ebert 6h ago', displays a table of build results for three applications: Shop, Design System, and Demo Studio. The bottom section, titled 'Christoph Ebert Friday', shows bundle changes for a commit, indicating a decrease in size of 30 B.

App	Preview	Built
Shop	https://galaxus-ch-preview-pr109497.preview.k8s.devinite.com	09:01 13.03.23
Design System	https://pr109497-dg-designsystem-host.preview.k8s.devinite.com	08:54 13.03.23
Demo Studio	https://pr109497-demo-studio.preview.k8s.devinite.com/demo-studio	08:52 13.03.23

Oliver Günthardt approved the pull request Friday

Oliver Günthardt joined as a reviewer Friday

Christoph Ebert Friday

Bundle Changes: - 30 B
Merge Commit: [a005b1f1](#) | Trigger Commit: [a759249a](#) | Master Build: [1269960](#) | PR Build: [1270183](#)

Chunk	Size	Diff
pages/after-sales	21.7 KB	- 30 B
chunks/pages/after-sales-df449909d09cd33ajs	21.7 KB	- 30 B



What are we talking about?

Server Side Rendering (SSR)

Server Side Rendering (SSR)

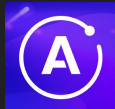
1. Client sends a request with a certain URL

Server Side Rendering (SSR)



2. We check if a cached document exists for the requested URL

Server Side Rendering (SSR)



3. Fetch initial data with `.getInitialProps()`

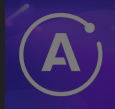
```
export interface HomePageProps {
  data: GetDailyDealPreviewsQuery;
}

export const HomePage: CustomPage<HomePageProps> = (props) => {
  return <Home {...props} />;
};

HomePage.getInitialProps = async ({
  apolloClient,
}: ICustomPageContext) => {
  const { data } = await apolloClient.grapholيث.query<
    GetDailyDealPreviewsQuery,
    GetDailyDealPreviewsQueryVariables
  >({
    query: GET_DAILY_DEAL_PREVIEWS,
  });

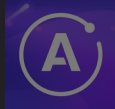
  return {
    data
  };
}
```

Server Side Rendering (SSR)



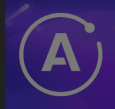
4. Render the document

Server Side Rendering (SSR)



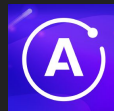
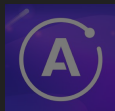
5. This includes Styled-Components

Server Side Rendering (SSR)



6. Done, right? 🤔

Server Side Rendering (SSR)

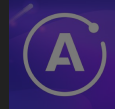
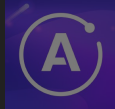


6. While rendering our components, we discover new components that are fetching data for themselves.

```
interface IProductDetailQuestionsProps {
  productId: string;
}

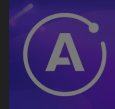
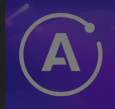
export const ProductDetailQuestions: FunctionComponent<
  IProductDetailQuestionsProps
> = ({ productId }) => {
  const { data, loading, error } = useAnsweredProductQuestionsQuery({
    variables: { productId },
  });
  if (error) {
    return (
      <AlertWrapper>
        <Alert variant="error">{__("Questions could not be loaded")}</Alert>
      </AlertWrapper>
    );
  }
  if (loading) {
    return (
      <LoadingCircleWrapper>
        <LoadingIndicatorCircle />
      </LoadingCircleWrapper>
    );
  }
  return data?.questions.items ? (
    <ProductDetailQuestionList
      title={__("The most helpful customer questions")}
      questionsAndAnswers={data.questions.items}
    />
  ) : null
};
```

Server Side Rendering (SSR)



7. Save the document in the Akamai cache, for the next user

Server Side Rendering (SSR)

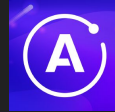
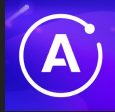


8. Send the document to the client



Lessons Learned

CDN Caching



The Server can't know about the user

CDN Caching

HTML Output Cache

- Great start for static content
- Lazy-load user specific content
- Limited personalization

Personalized Content

The screenshot shows the GALAXUS website interface. At the top, there's a search bar and user navigation icons. The main content area features a large image of the Samsung Odyssey Ark monitor. Below the image, the product title is 'Zu viel für mich: Samsungs Monster-Monitor Odyssey Ark im Test'. The price is listed as 2399,-. The page includes a detailed description of the monitor's features, such as its 55-inch curved display and 1000R curvature. On the right side, there are promotional banners for 'Galaxus Tagesangebot' and 'Galaxus Live'. A red arrow points to the user profile icon in the top right corner, with the text 'Personalized Content' next to it.

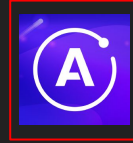
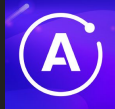
💡 W/O CDN Caching

- Optimize your Infrastructure & APIs
- Avoid Data Fetching Waterfalls
- API-Level Caching



DATADOG

Data-Fetching in Components



Components fetching their own data leads to waterfalls in rendering and data loading




Data fetching waterfall

galaxus.ch/de/s1/product/apple-iphone-13-mini-128-gb-midnight-540-sm-esim-12-mpx-5g-smartphone-166447867ip=iphone+

GALAXUS

Suche nach Produkten, Kategorien, Marken und mehr

Gesamtsortiment > IT + Multimedia > Smartphones + Tablets > Smartphone > Apple iPhone 13 mini



624.97 statt 659,-
Apple iPhone 13 mini
128 GB, Midnight, 5.40", SIM + eSIM, 12 Mpx, 5G

Bewertungen
★★★★★ 651

Testberichte
Sehr gut bei 10 Tests

Morgen geliefert
Mehr als 10 Stück in unserem Lager

In den Warenkorb

Vergleichen Merken

Speicherkapazität

128 GB 256 GB 512 GB

Farbvariante

Midnight (PRODUCT)RED Blue Green Pink Starlight

Elements Console Performance insights Sources Network Performance Memory Application Lighthouse AdBlock Piez

Preserve log Disable cache No throttling

/api/graphql

Name	Status	Domain	Type	Initiator	Size	Time	Priority	C...	Waterfall
get-recently-visited-products	200	www.galaxus.ch	fetch	app-5362e68208ab4a5.js	1.3 kB	186 ms	High	br	
get-recently-visited-products	200	www.galaxus.ch	fetch	app-5362e68208ab4a5.js	1.3 kB	217 ms	High	br	
get-product-list-counts	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	981 B	199 ms	High	br	
get-product-details-structured-data	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	2.1 kB	197 ms	High	br	
get-main-navigation-desktop-v2	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	1.6 kB	398 ms	High	br	
get-offer-availability	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	1.4 kB	220 ms	High	br	
pdp-get-warranties-details	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	1.3 kB	533 ms	High	br	
get-avatar	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	1.0 kB	600 ms	High	br	
get-user-menu-pending-items-count	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	979 B	691 ms	High	br	
get-flash-delivery-availability	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	1.0 kB	690 ms	High	br	
track-product-detail-pageview	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	933 B	598 ms	High	br	
pdp-get-product-details-critical-data-refetch	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	2.9 kB	624 ms	High	br	
get-cart	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	1.1 kB	615 ms	High	br	
track-pageview	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	919 B	536 ms	High	br	
pdp-co2-compensation	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	1.1 kB	503 ms	High	br	
pdp-get-comparison-product-details-mini-specifications	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	1.7 kB	477 ms	High	br	
pdp-get-marketing-page-ids-by-product-id	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	930 B	518 ms	High	br	
answered-product-questions	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	2.6 kB	417 ms	High	br	
unanswered-product-questions	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	960 B	417 ms	High	br	
get-community-rating-of-product-by-user	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	919 B	460 ms	High	br	
get-product-rating-overview	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	1.0 kB	426 ms	High	br	
get-products-with-offer-default	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	2.0 kB	425 ms	High	br	
get-main-navigation-mobile-v2	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	1.7 kB	446 ms	High	br	
get-offer-availability	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	1.4 kB	275 ms	High	br	
get-products-with-offer-default	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	6.7 kB	256 ms	High	br	
pdp-co2-compensation	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	1.1 kB	123 ms	High	br	
get-products-with-offer-default	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	2.3 kB	81 ms	High	br	
pdp-get-marketing-page-previews	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	1.2 kB	163 ms	High	br	
get-user-votetype	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	1.1 kB	162 ms	High	br	
get-products-with-offer-default	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	2.2 kB	163 ms	High	br	
get-most-helpful-reviews	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	934 B	177 ms	High	br	
get-community-ratings	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	4.9 kB	170 ms	High	br	
get-products-with-offer-default	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	2.5 kB	76 ms	High	br	
get-recently-visited	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	1.1 kB	101 ms	High	br	
get-products-with-offer-default	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	966 B	75 ms	High	br	
get-products-with-offer-default	200	www.galaxus.ch	xhr	app-5362e68208ab4a5.js	7.8 kB	12 ms	High	br	

DON'T DO THAT



The Power of GraphQL Fragments

```
// ProductPage.tsx
import { DocumentType, gql } from "@segments/graphql";
import { CustomPage, ICustomPageContext } from "@segments/page";
import { ProductDetail } from "../ProductDetail";

const GetProductPageDocument = gql(/* GraphQL */ `
  query GetProductPage($id: ID!) {
    shopProduct(id: $id) {
      ...ProductDetail_ShopProduct
    }
  }
`);

interface IProductPageProps {
  data: DocumentType<typeof GetProductPageDocument>;
}

export const ProductPage: CustomPage<IProductPageProps> = ({ data }) => {
  return <ProductDetail shopProduct={data.shopProduct} />;
};

ProductPage.getInitialProps = async ({ apolloClient,
ICustomPageContext }) => {
  const { data } = await apolloClient.supergraph.query({
    query: GetProductPageDocument,
    variables: {
      id: "123",
    },
  });
};

return {
  data,
};
};
```

```
// ProductDetail.tsx
import { FragmentType, gql, unmaskFragment } from
"@segments/graphql";
import { FunctionComponent } from "react";
import { ProductImages } from "../ProductImages";

const ProductDetail_ShopProduct = gql(/* GraphQL */ `
  fragment ProductDetail_ShopProduct on ShopProduct {
    name
    ...ProductImages_ShopProduct
  }
`);

interface IProductDetailProps {
  shopProduct: FragmentType<typeof ProductDetail_ShopProduct>;
}

export const ProductDetail: FunctionComponent<IProductDetailProps> =
({ shopProduct: shopProductData,
}) => {
  const shopProduct = unmaskFragment(
    ProductDetail_ShopProduct,
    shopProductData
  );

  return (
    <
      <h1>{shopProduct.name}</h1>
      <ProductImages shopProduct={shopProduct} />
    </>
  );
};
```

```
// ProductImages.tsx
import { FragmentType, gql, unmaskFragment } from
"@segments/graphql";
import { FunctionComponent } from "react";

const ProductImages_ShopProduct = gql(/* GraphQL */ `
  fragment ProductImages_ShopProduct on ShopProduct {
    images {
      url
      description
    }
  }
`);

interface IProductImagesProps {
  shopProduct: FragmentType<typeof ProductImages_ShopProduct>;
}

export const ProductImages: FunctionComponent<IProductImagesProps> =
({ shopProduct: shopProductData,
}) => {
  const shopProduct = unmaskFragment(
    ProductImages_ShopProduct,
    shopProductData
  );

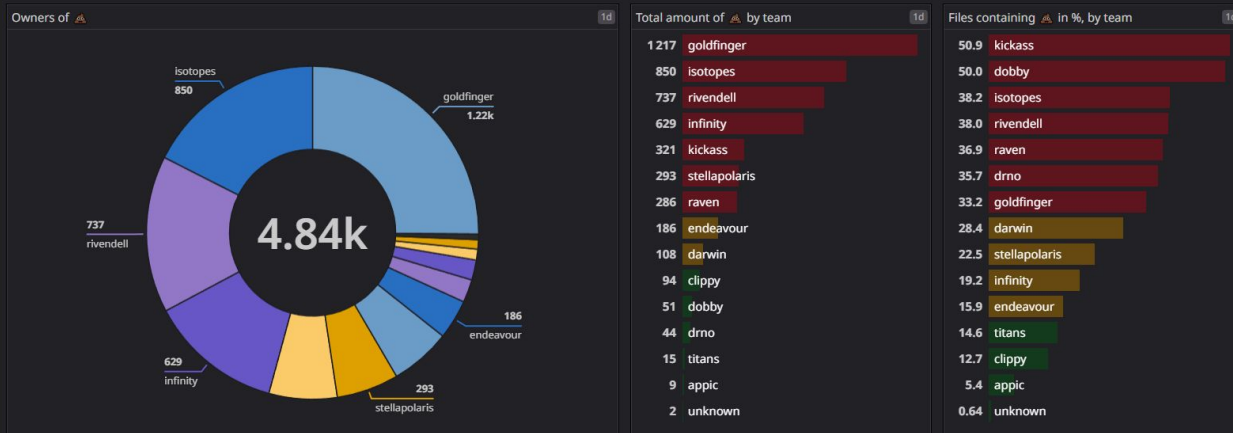
  if (!shopProduct.images?.length) {
    return null;
  }

  return (
    <ul>
      {shopProduct.images.map((image) => (
        <li key={image.url}>
          <img src={image.url} alt={image.description} />
        </li>
      ))}
    </ul>
  );
};
```

Final Thoughts

📖 Fight Technical Debt

- Measure it!
- Continuous Refactoring (Days)





Thanks

Questions, Feedback, Thoughts?



Like to join?
galaxus.ch/joboffer