# RODENSTOCK

**Rodenstock CNXT – a new data integration platform**

Tom Weber

London, OSVA Conference, 8th May 2019

# AGENDA

1. Background & Motivation
2. Measurement Equipment Integration
3. Rodenstock CNXT - a new data integration platform
4. Rodenstock CNXT API demo

# Background

- More and more customers (opticians) use mobile devices for their daily business

- Rodenstock needs to develop new applications as modern (progressive) web application for targeting multiple platforms (Microsoft Windows, Mac OS X, iOS, or Android)

- Gartner said that more than 50% of mobile apps will be implemented as Progressive Web Application (PWA) by 2020 (see https://www.gartner.com/en/documents/3645344)

# Technical briefing: Progressive Web Application (PWA)

- Overcome the potential limitations of the web such as performance, quality, and unavailable Internet connection
- Provides a smooth, app-like experiences for the mobile web

- Disrupts the mobile app paradigm by bridging the web experience with native app functionality

To sum it up:

- Installable
- Lightweight & fast
- App-like and responsive
- Connectivity independent
- Secure

# Background

**Feedback from our customers:**

- Missing data integration into 3rd party applications for ImpressionIST® 4
- No consistent and no intuitive workflows for data exchange between 3rd party applications and Rodenstock measurement devices
- More than 50% of (german) opticians are dissatisfied with the current data integration between 3rd party applications and Rodenstock measurement devices
- 88% of opticians in Germany use an iPad and 6% use an Android tablet for working with the ImpressionIST® 4

→ Opticians want a more convenient solution provided by Rodenstock and branch software manufacturers

# Motivation

- Establish an API based data exchange instead of a “legacy” file system based one
- Improve user (optician) experience
- Reduce development costs (one team for targeting all platforms)
- Reduce test, deployment, and maintenance effort
- Reduce time to market for new features, bugfixes etc.

# Goals

- Rodenstock will provide and publish an convenient Open API for integrating data from Rodenstock measurement devices into 3rd party applications

- Rodenstock will support import and export of B2BOptic lens order XSD Version 1.6.3 and future versions

# Why is an API approach best at present?

- APIs are the de-facto standard for developing and connecting modern (web-) applications

- APIs and API management have become essential to how enterprises deliver applications in and across the web or clouds

# AGENDA

1. Background & Motivation
2. Measurement Equipment Integration
3. Rodenstock CNXT - a new data integration platform
4. Rodenstock CNXT API demo

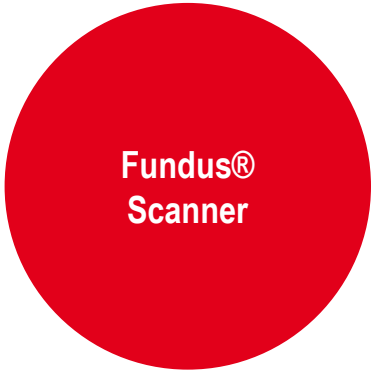
# Rodenstock Measurement Equipment



# Measurement Equipment Integration (MEI)

Available measurement devices:






Supported interfaces:

SOAP based API with custom XML Schema Definition

File system based (B2BOptic XSD)

File system based (IOPCFG XSD)

Planned interfaces:

RESTful or GraphQL API provided by CNXT Connector

RESTful or GraphQL API provided by CNXT Connector

RESTful or GraphQL API provided by CNXT Connector

# API main use cases

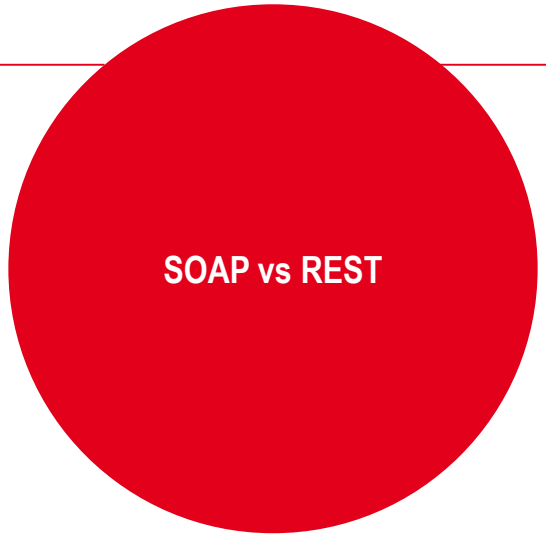
- Query patients defined by last name, first name, or date of birth
- Query sessions of a patient defined by patient id, session date, or session name
- Import patient related data from 3rd party applications
- Export patient related data to 3rd party applications

# SOAP vs REST

## SOAP

Commonly used in enterprise environments but currently not in modern web apps

+XML Schema Definition (type control and schema validation)
- XML tag overhead
- Cannot be fetched in web browser without programmatically parsed message body




## REST

Commonly used for exposing Open APIs

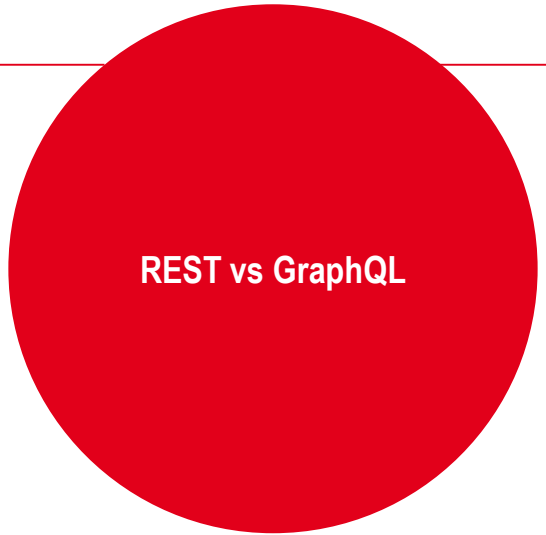
+REST is generally faster and uses less bandwidth
+REST and JSON has become the de-facto technology for the majority of Open APIs
+Data fetching natively supported in web browsers
- No type control

# REST vs GraphQL

## REST

An architectural principle competing with SOAP

+Easy to consume in modern web apps
- No type control
- Multiple endpoints (for every resource)
- Over- and under-fetching




## GraphQL

A methodology competing with REST

+Supports a strong type system via GraphQL schema
+GraphQL Schema serves a contract between client and server
+Perfect for exposing complex data
+One endpoint for all resources
+Data fetching natively supported in web browsers

# Technical briefing: Representational Stateful Transfer (REST)

Typically used in web application for realising a CRUD szenario:

CREATE – HTTP POST

READ – HTTP GET

UPDATE – HTTP PUT / PATCH

DELETE – HTTP DELETE

# Technical briefing: GraphQL

Queries

- Used by the client to request the data it needs from the server

Mutations

- Used by the client for CUD (Create, Update, Delete)
- Syntax for mutations look almost the same as queries, but they must start with the mutation keyword

Subscriptions

- Create and maintain real time connection to the server via web sockets
- Enables the client to get immediate information about related events

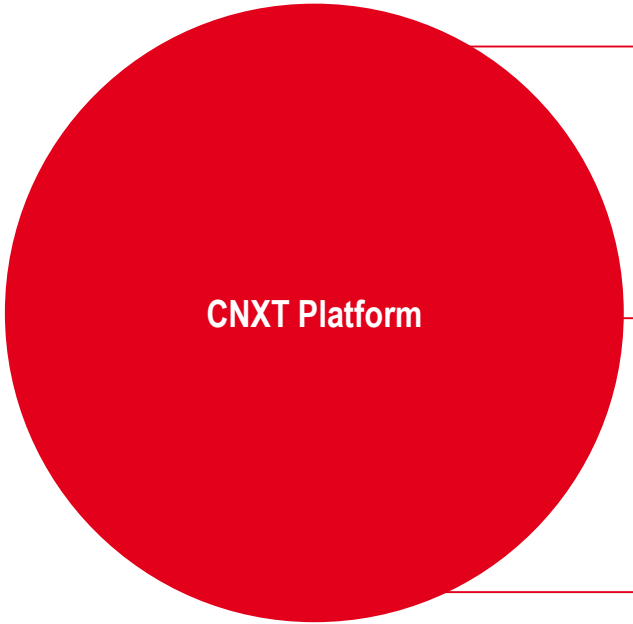
# AGENDA

1. Background & Motivation
2. Measurement Equipment Integration
3. Rodenstock CNXT - a new data integration platform
4. Rodenstock CNXT API demo

# Rodenstock CNXT - a new data integration platform

- API based data integration platform
- Provides an Open API designed and published by Rodenstock
- Provides a RESTful and a GraphQL based API for integrating data into 3rd party applications
- Provides MQTT based messaging reliability and persistence
- Provides Single Sign-On (SSO) authentication and authorization

# Architectural overview and components




## CNXT Hub

- Centralized backend which handles complex business logic for identifying patient related data and merging B2BOptic XML documents
- Provides consulting and order related data

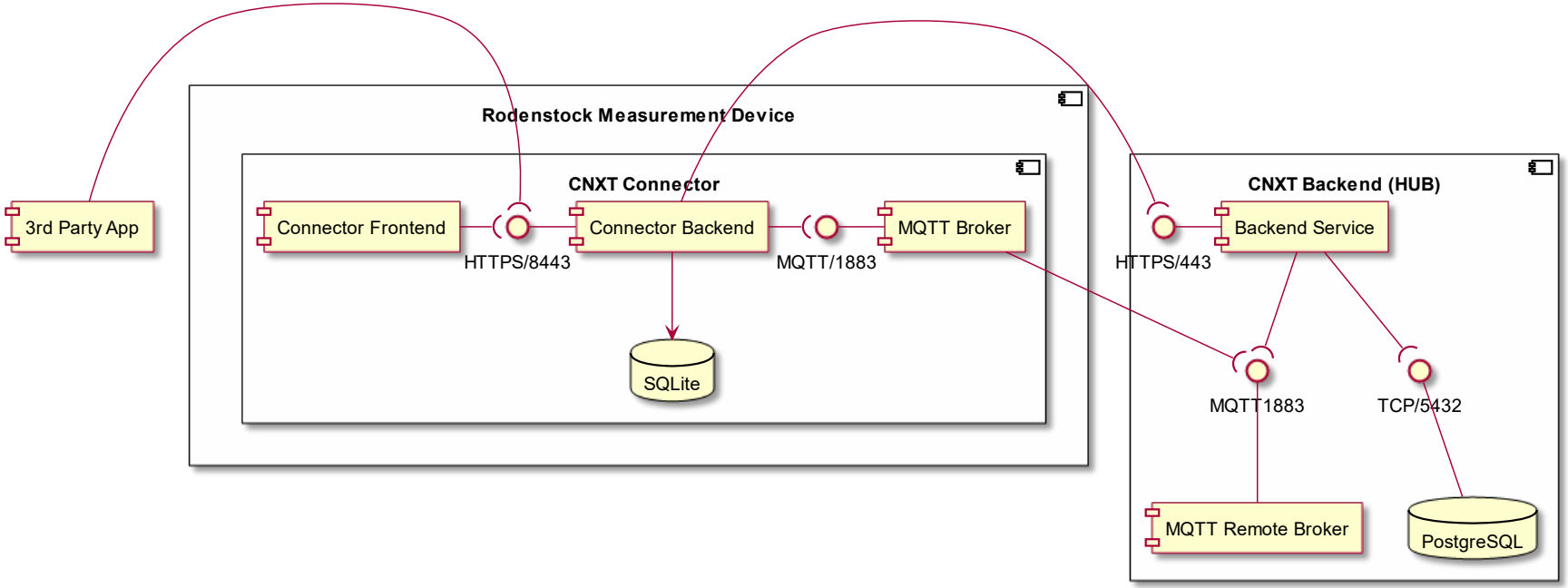
## CNXT Connector

- Locally installed app based on Electron / NodeJS
- Provides an Open API for integrating data from measurement devices or from 3rd party applications

## MQTT Broker

- Provides a lightweight method for carrying out messaging using a publish/subscribe mode
- Guarantees Quality of Service (QoS)
- Coupling of IoT devices

# Component diagram



# AGENDA

1. Background & Motivation
2. Measurement Equipment Integration
3. Rodenstock CNXT - a new data integration platform
4. Rodenstock CNXT API demo

# Rodenstock CNXT API demo

## RESTful endpoint

- Available via http://backend.external.cnxt.dtr01.rodenstock.com:8080

- The API specification is made with Swagger-UI: (see http://backend.external.cnxt.dtr01.rodenstock.com:8080/swagger-ui.html)

## GraphQL endpoint

- Available via http://backend.external.cnxt.dtr01.rodenstock.com:8080/graphql

- A visualization of the GraphQL schema is available via http://backend.external.cnxt.dtr01.rodenstock.com:8080/voyager

# Thank you for your attention.

RODENSTOCK