



Onboarding

Davra Platform Quick Start Guide

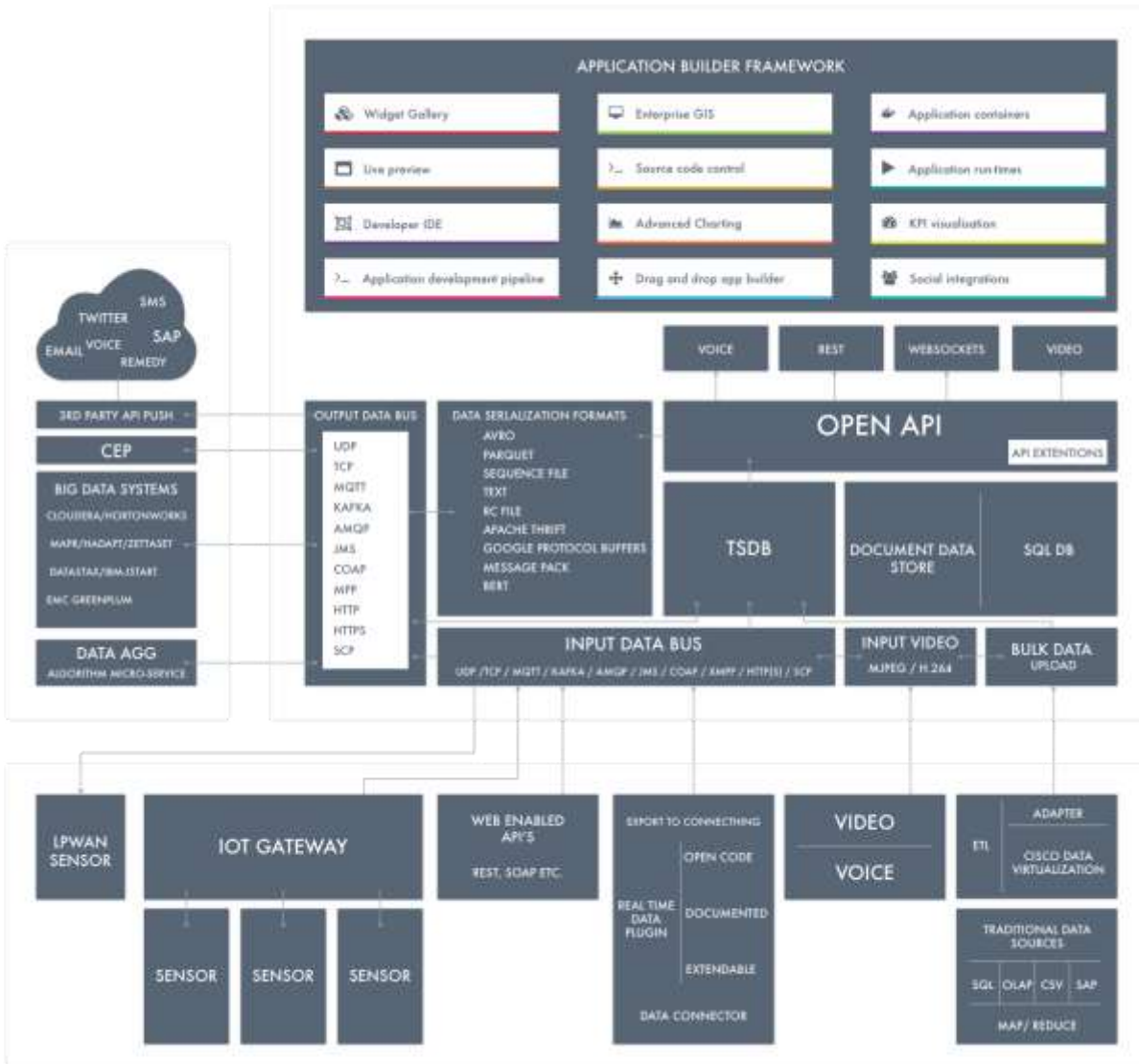
20th Feb 2020

Contents

INTRODUCTION	3
INSTALLATION TYPES	4
ACCESS THE USER INTERFACE	5
ADD DEVICE	8
CONFIGURE RULES	12
BUILD AN APP	13
GIS – GEO-TRACKING AND GEO-FENCING	24
ADDITIONAL FEATURES	26

Introduction

Davra is a platform with the capabilities to track and manage large deployments of devices and sensors, while also enabling technical customers to multiply their return on engineering resources by accelerating the speed of development. In other words, Davra is both an IoT and an AEP platform and this document describes the end-to-end process for installing the platform, adding devices, collecting and processing data, and building custom applications. An overview of the Davra architecture is shown below.



Installation Types

There are three primary ways to have the Davra platform installed and hosted: on premise, public cloud and private cloud.

On-Prem Install

An on-prem installation is set up using the customer's own resources for hosting the Davra platform on a virtual or physical machine. A CoreOS image is deployed and configured as a host OS and Davra is installed on top. The Davra knowledge base contains [detailed instructions](#) for the CoreOS and Davra installation process and Davra offers maintenance contracts for customers lacking the in-house resources to manage the deployments long-term.

Public Cloud

Public cloud instances are hosted on the AWS infrastructure and can be ordered via the AWS Marketplace or directly from Davra. This option allows customers to get a Davra instance up and running very quickly and reduces up-front costs for installation and configuration as well as long-term maintenance costs. To order a public cloud instance from Davra directly, please email sales@davra.com.

Private Cloud (Contact us)

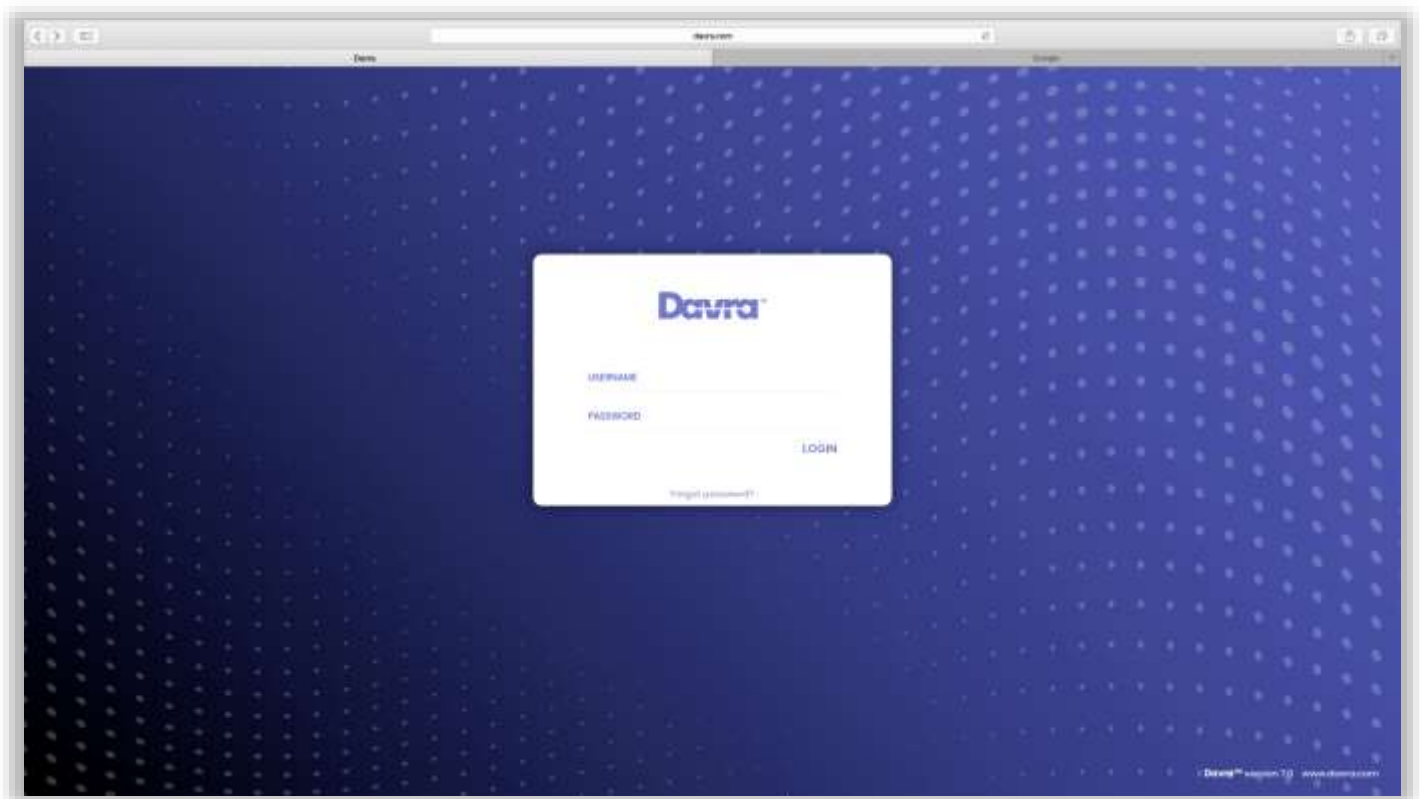
A private cloud is a multi-tenant Davra deployment hosted on the customer's own infrastructure that allows the customer to easily set up large numbers of independent instances, or tenants. This option is in most cases chosen by technical partners which host and manage Davra for their own customers. To discuss the implementation of a private cloud, please email sales@davra.com.

Access the User Interface

Once installed the Davra UI can normally be accessed at port 80 or 58000 at the appropriate IP address or URL, assuming that any firewalls between the user and the Davra server have been properly configured.

Log in

The login page is accessible at the platform URL after install over http or https.

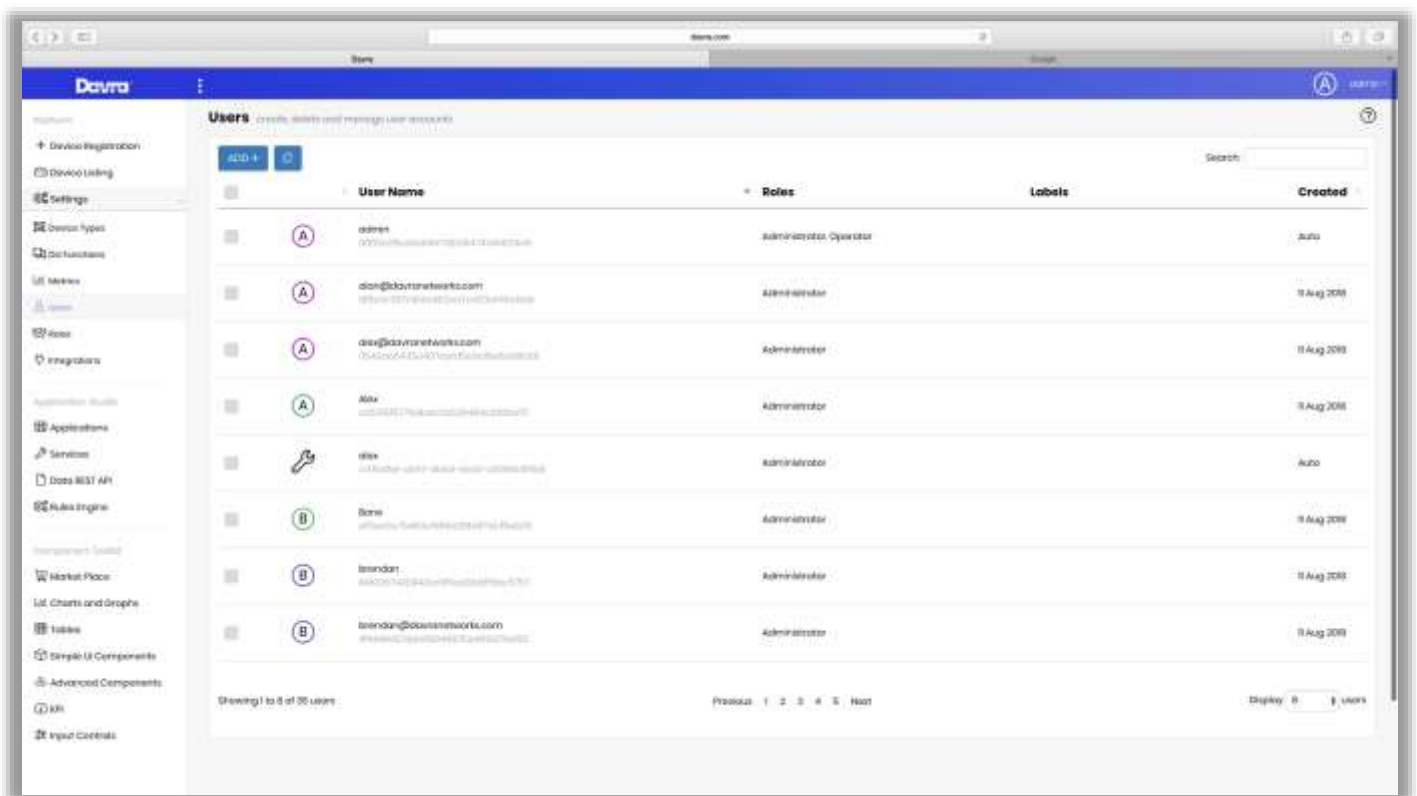


Login Screen

After install the Davra user interface can be accessed using either the authentication details provided by Davra or the default login details as per the Davra knowledge base.

User Roles

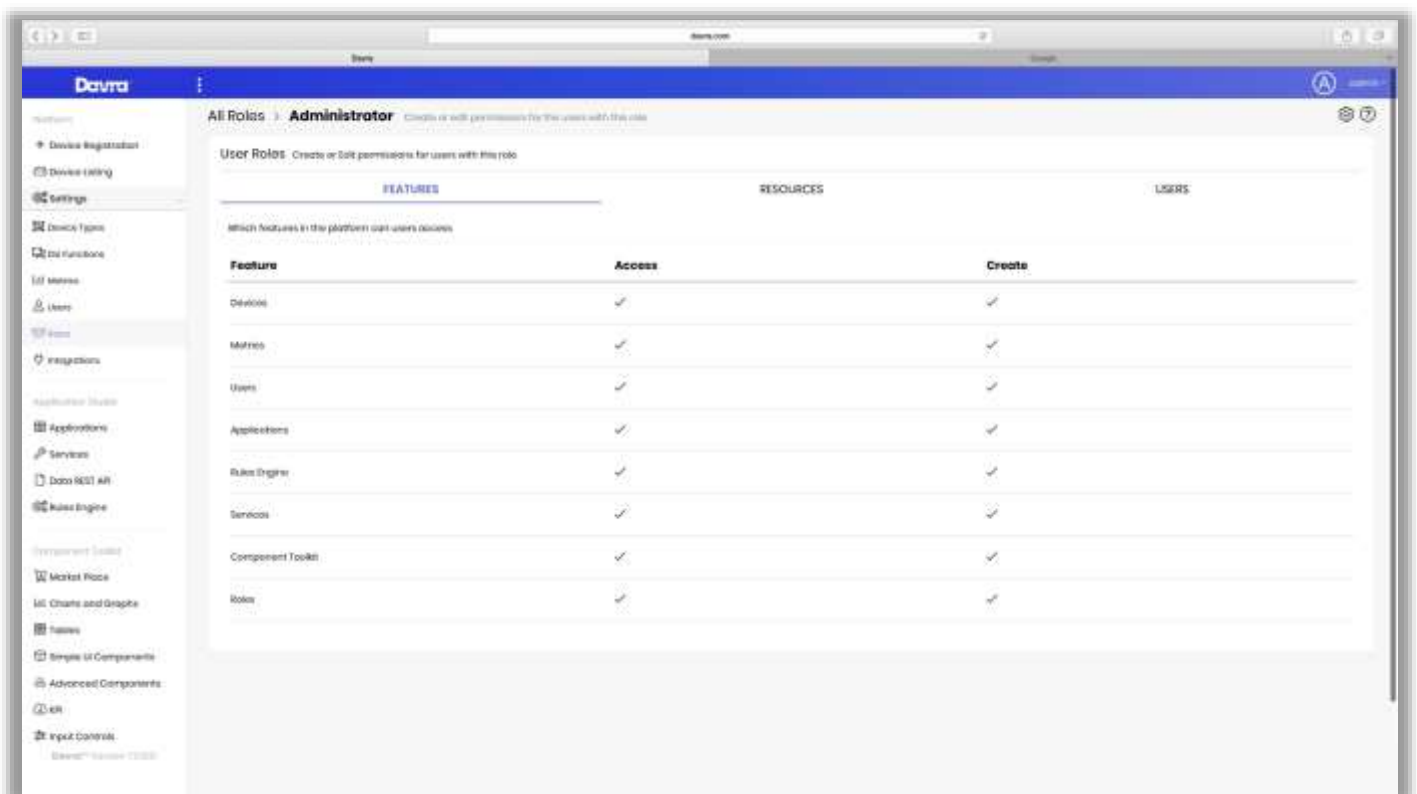
Users can be assigned administrator or operator roles depending on the nature of their intended work.



User Roles Admin

User Permissions

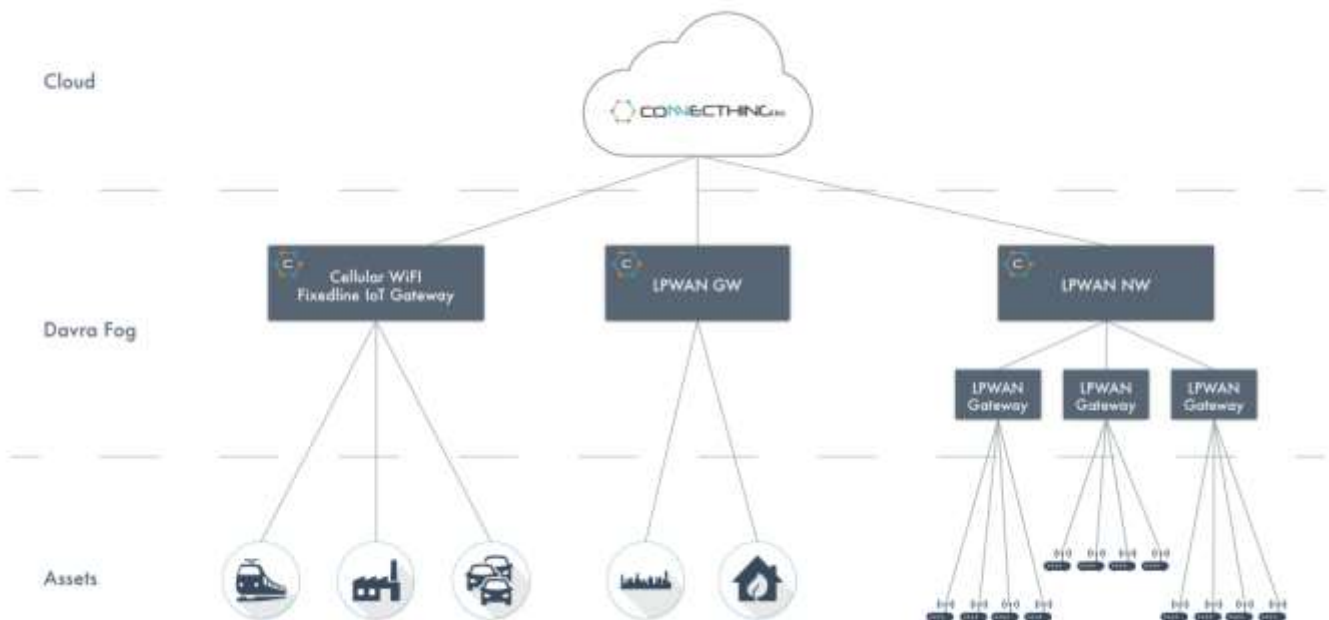
Users can be assigned various degrees of access to applications and devices, allowing for a secure and easy-to-use UI for all users.



User Permissions

Add Device

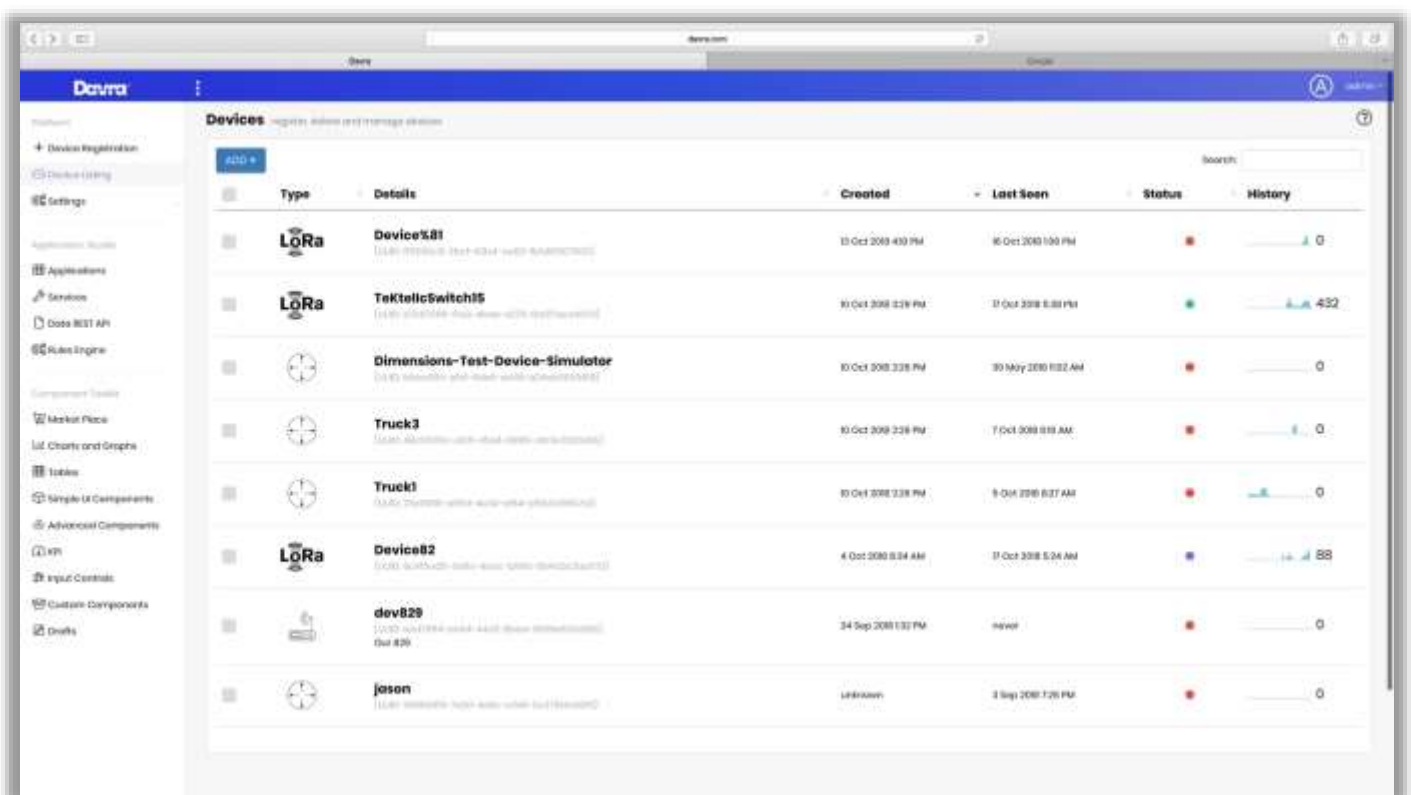
An IoT project normally includes a number of devices types, including sensors, gateways and network servers.



IOT and Lora Gateways

Typically, the IOT data sensors are connected to a gateway such as a router or a Lora gateway, feeding sensor data into the Davra platform. The gateways may also have more advanced agents installed performing device-side data processing or local caching, enabling them to deal with a disruption in connectivity with no data loss, and also keeping their ability to perform logical tasks while offline.

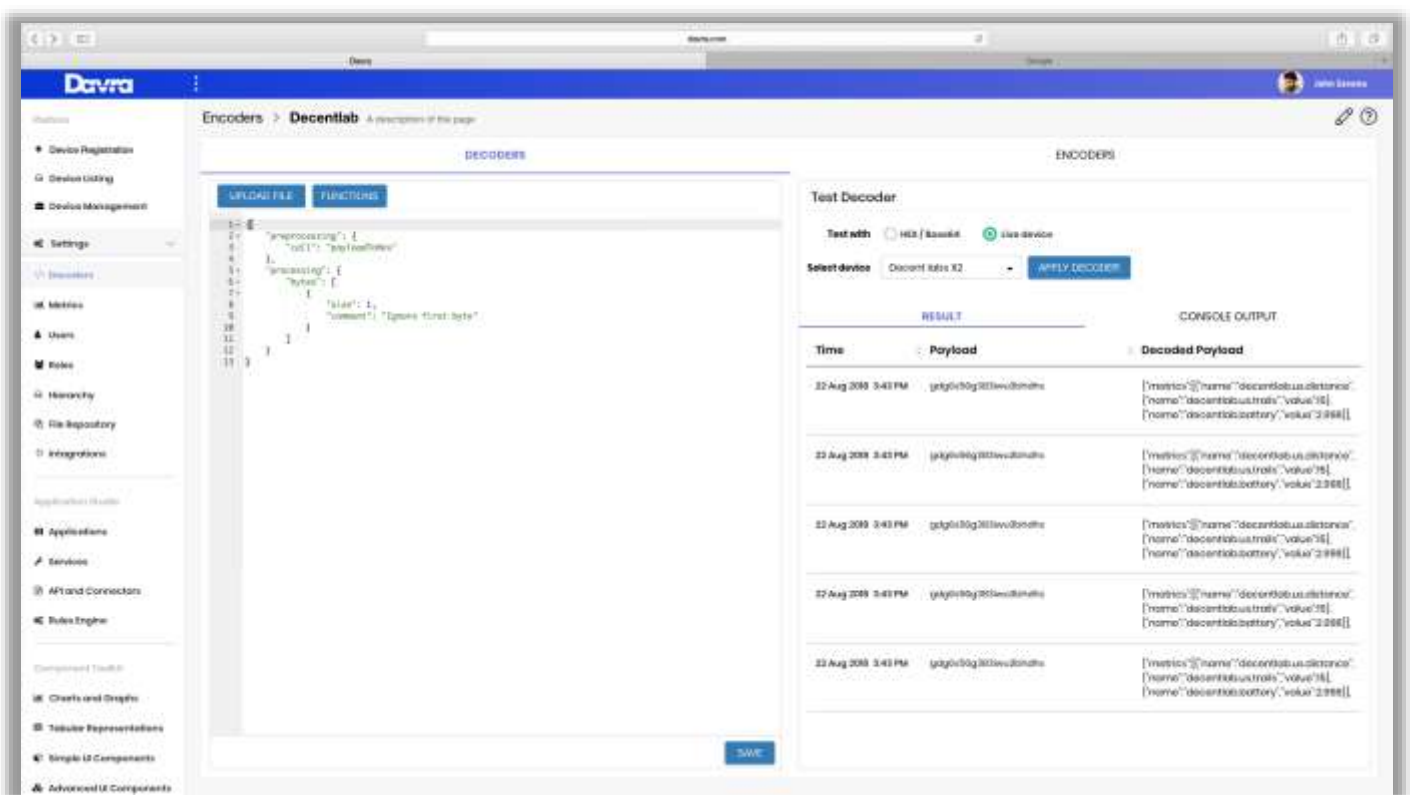
For Lora deployments network servers such as Orbiwise, Actility or Stream are often part of the solution, forwarding sensor data from the gateways to the Davra server. Davra has built-in support for integration with the most common network servers.



Device Listings

Data Decoders

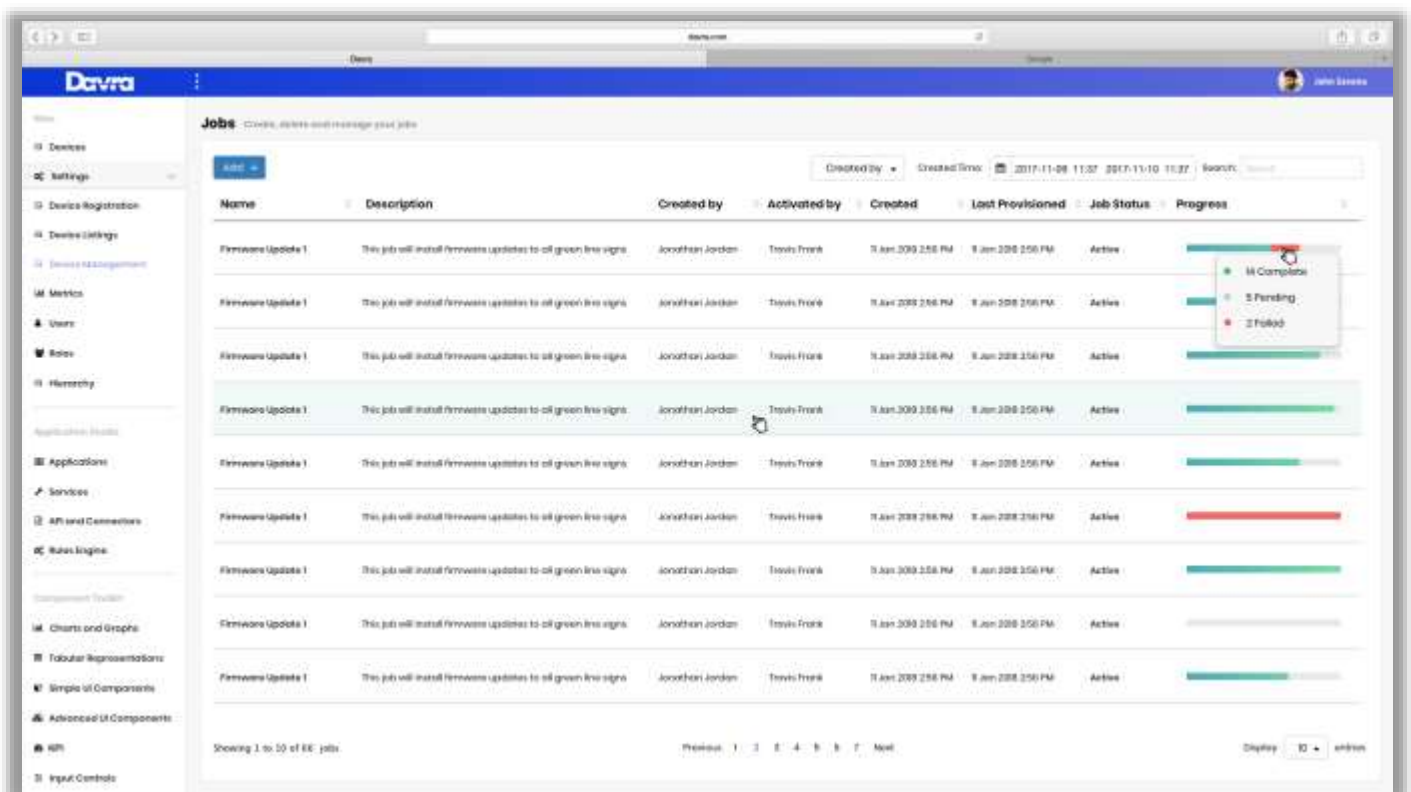
Davra comes with a number of payload decoders for Lora hex payloads and a development environment for creating additional decoders quickly and easily.



Data Decoders

Provisioning Process

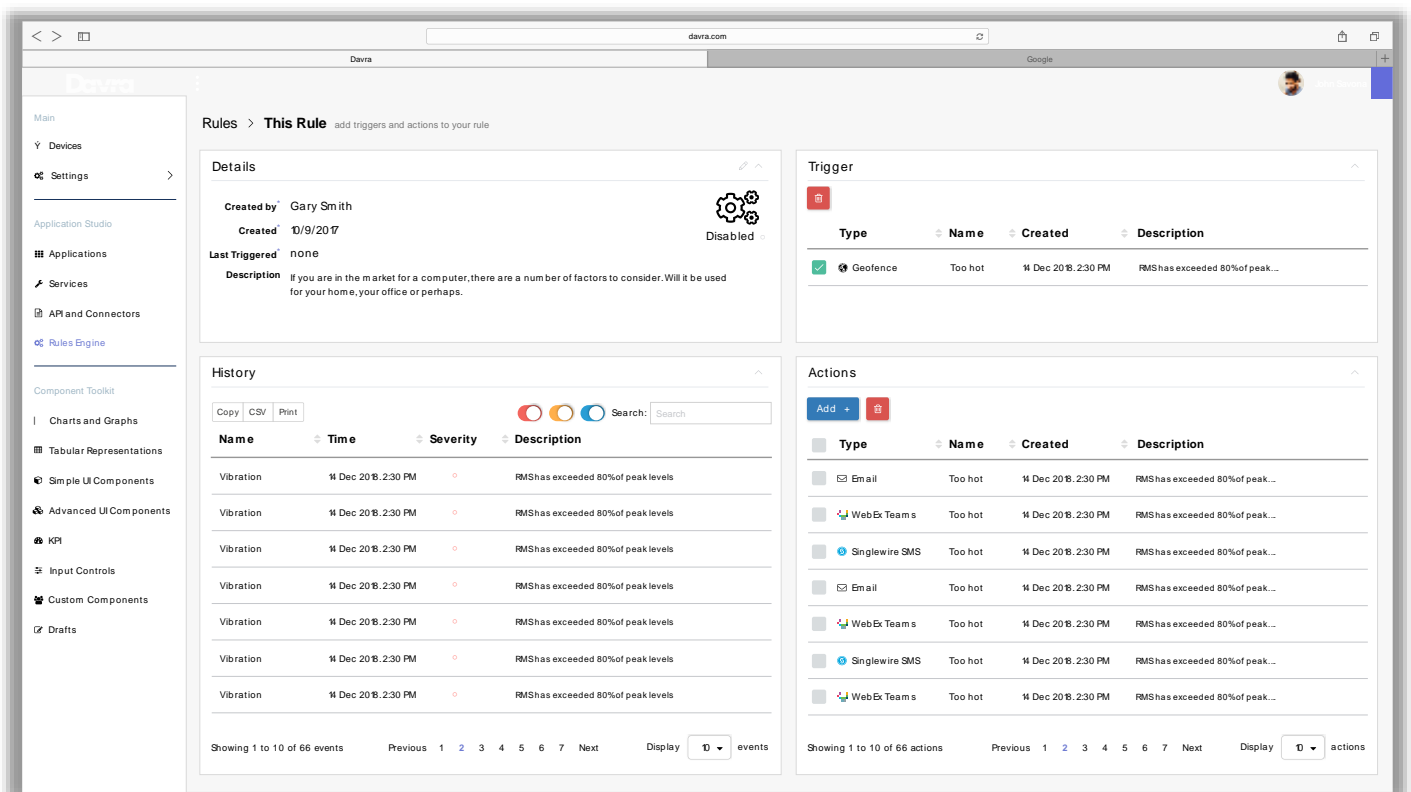
The device provisioning process consists of initial gateway-side pre-deployment configuration, enabling the devices point to the Davra server once they gain connectivity. This can be done by Davra or a technical partner. Once the devices have been provisioned the device-side configuration can be updated on the server-side so as to efficiently manage any number of devices in terms of device-side logic, data decoders and general configuration.



Zero Touch Provisioning

Configure Rules

The rules engine allows for alerts and actions to be triggered based on a set of configurable logical rules. For example, a rule configured for fleet management purposes might detect a speeding event and generate an email alert, or a rule for smart agriculture might detect a particular combination of moisture and temperature and trigger a sprinkler to turn on.



Rule Settings

Build an App

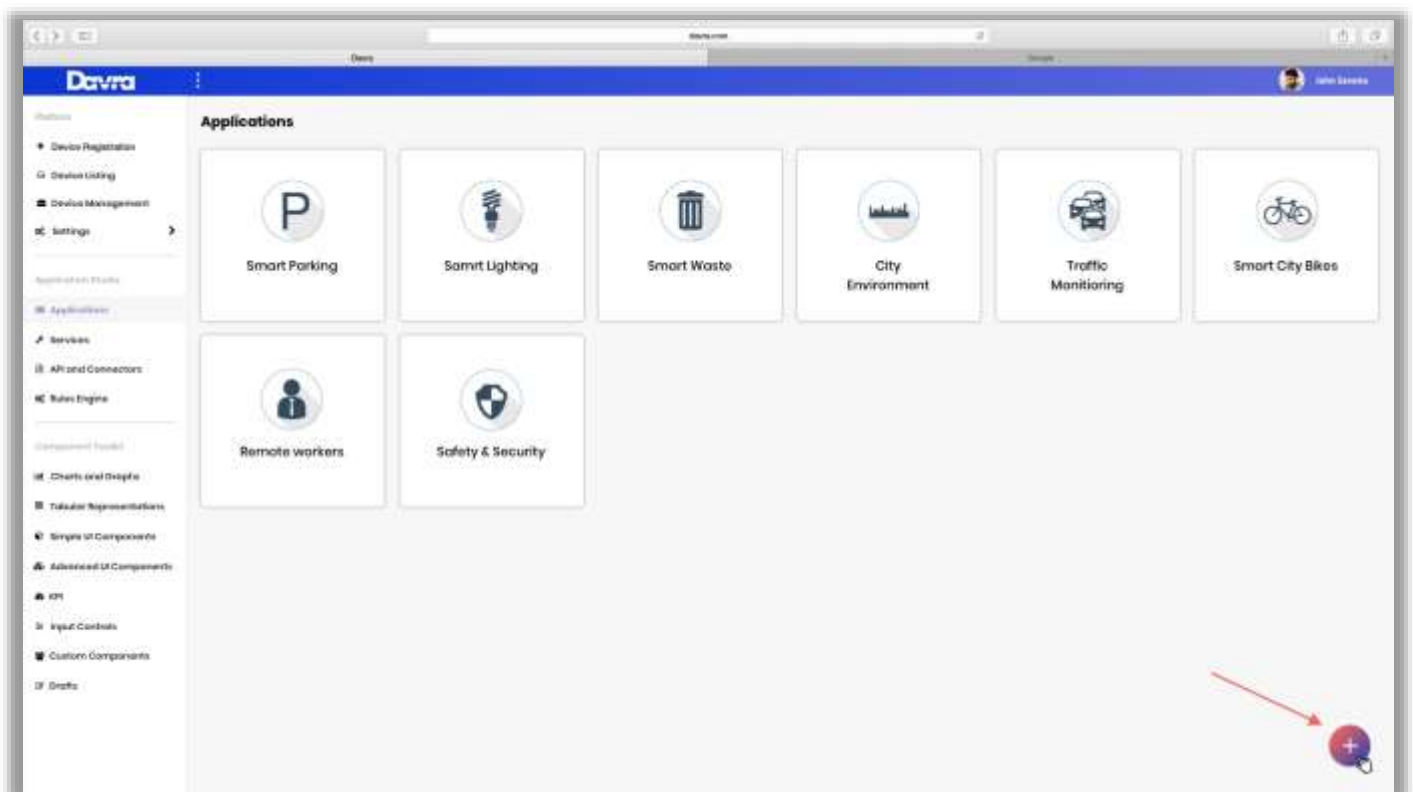
Davra is a full-fledged AEP that allows for rapid prototyping as well as time-efficient development of production-ready applications and integration with external systems. The AEP capabilities consist of two main parts:

- A front-end / UI development environment.
- A back-end / microservices development environment

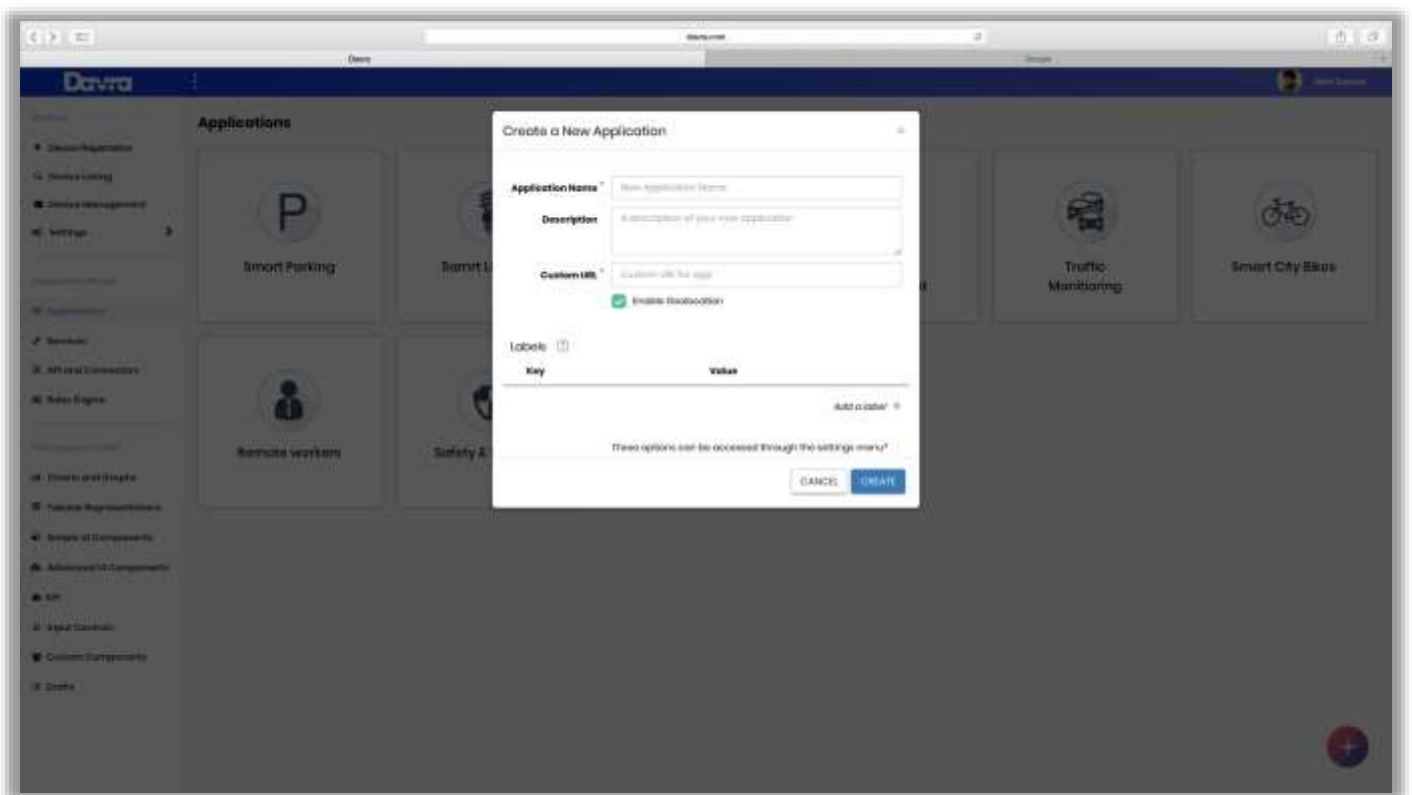
In addition to the development environments Davra also comes with a Git-based build and deployment system.

Add New App

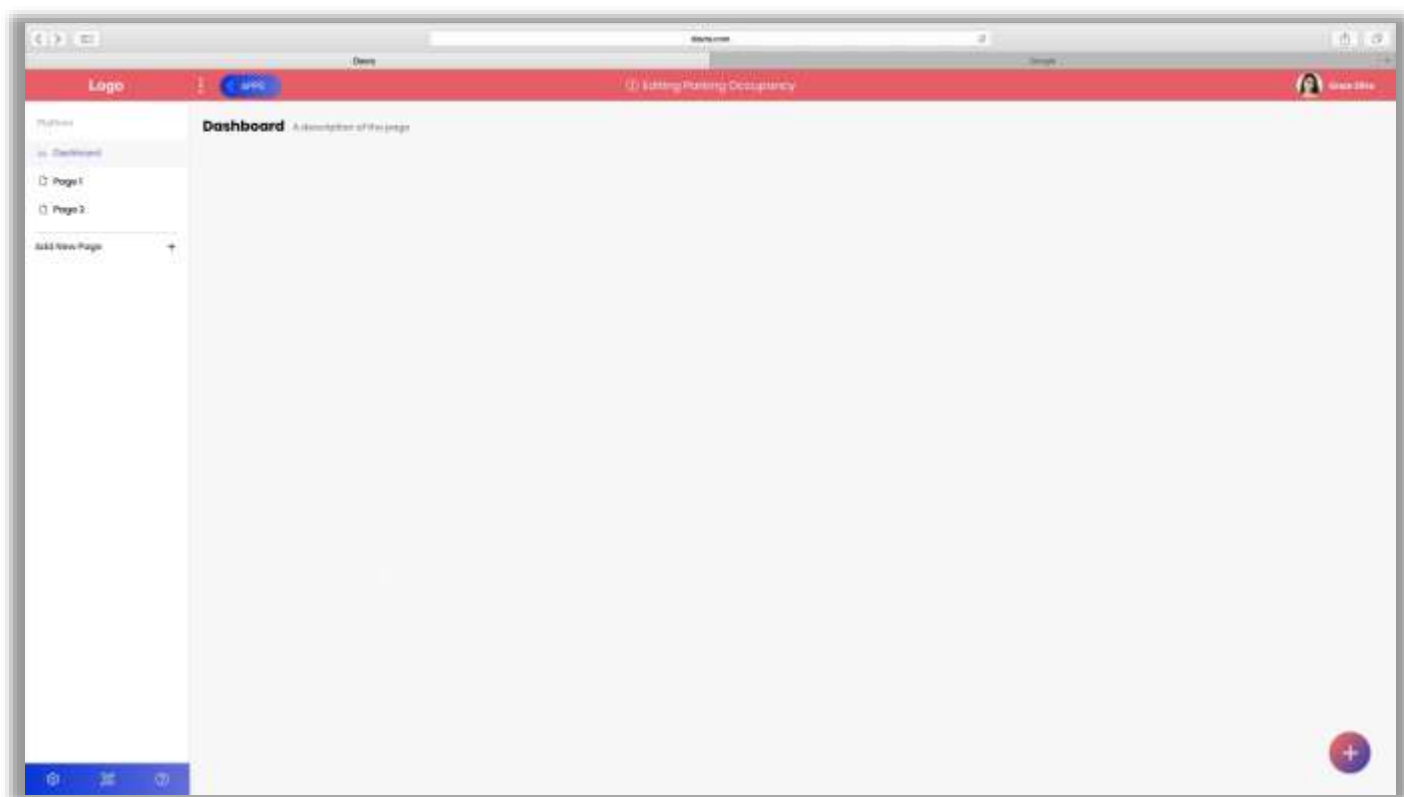
From the start page in Davra the user can easily create a new app which can contain multiple pages and widgets.



Create a new App



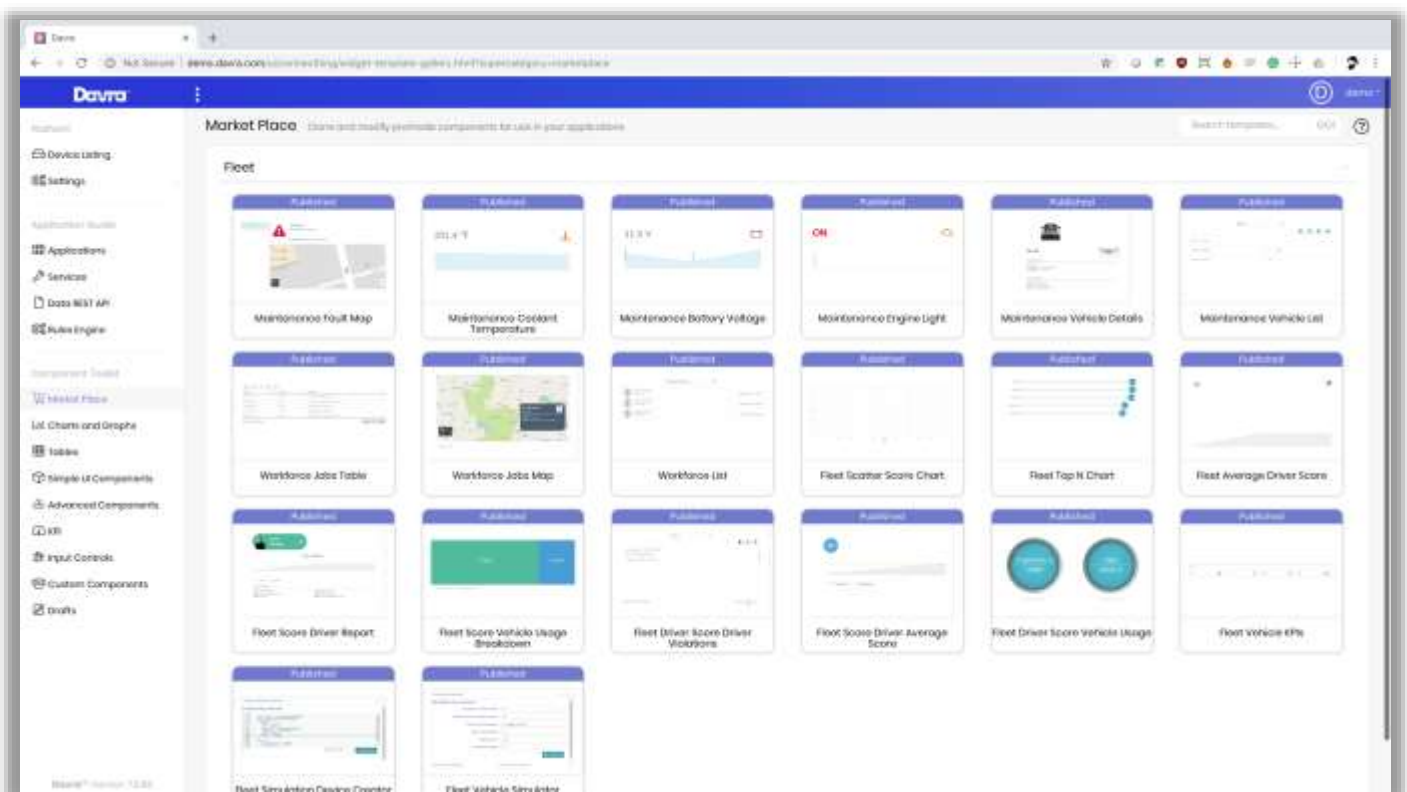
Enter new App details



New App Landing Page

Frontend – Built-in Widgets

Davra comes with an extensive gallery of built-in components including line charts, bar charts, tables, gauges, KPI indicators and more.



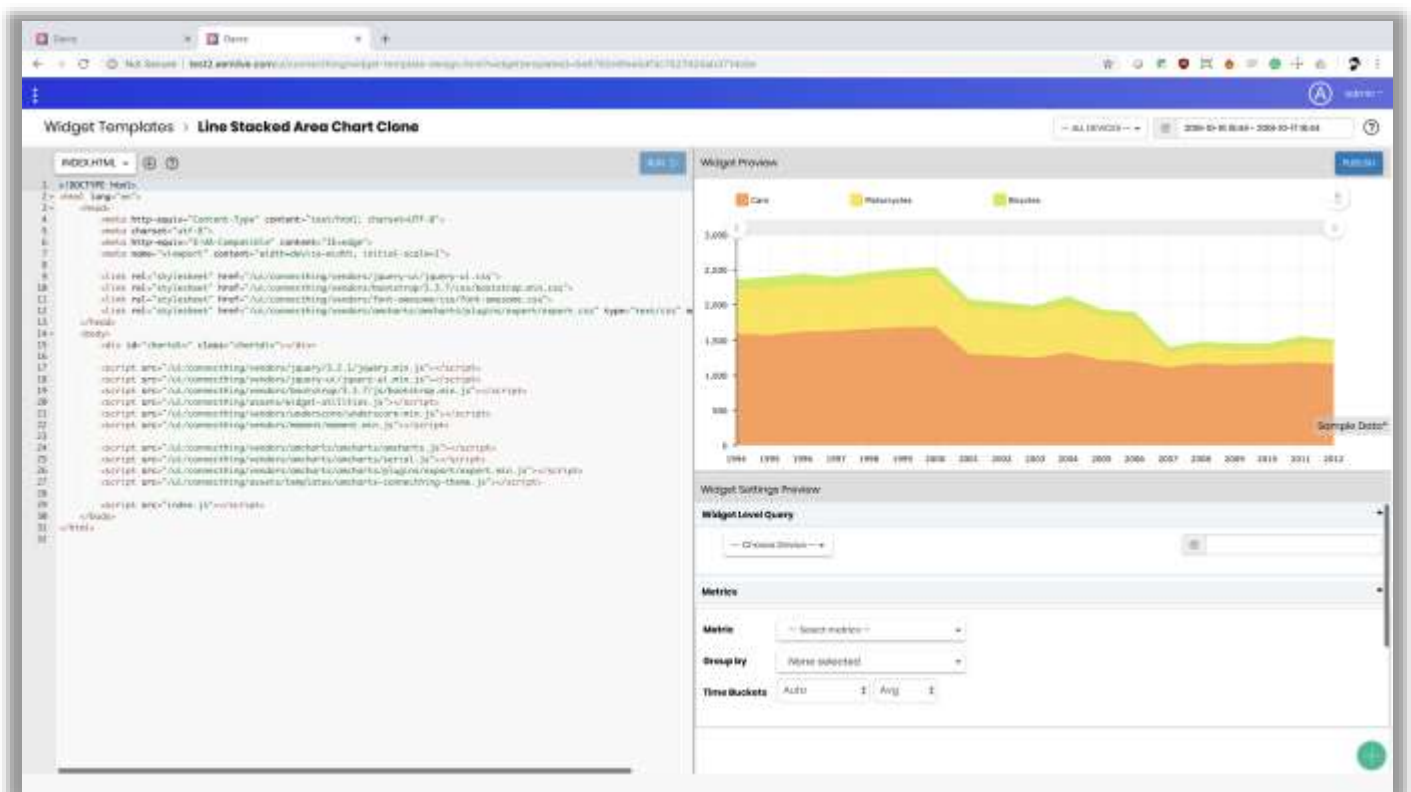
Component Toolkit



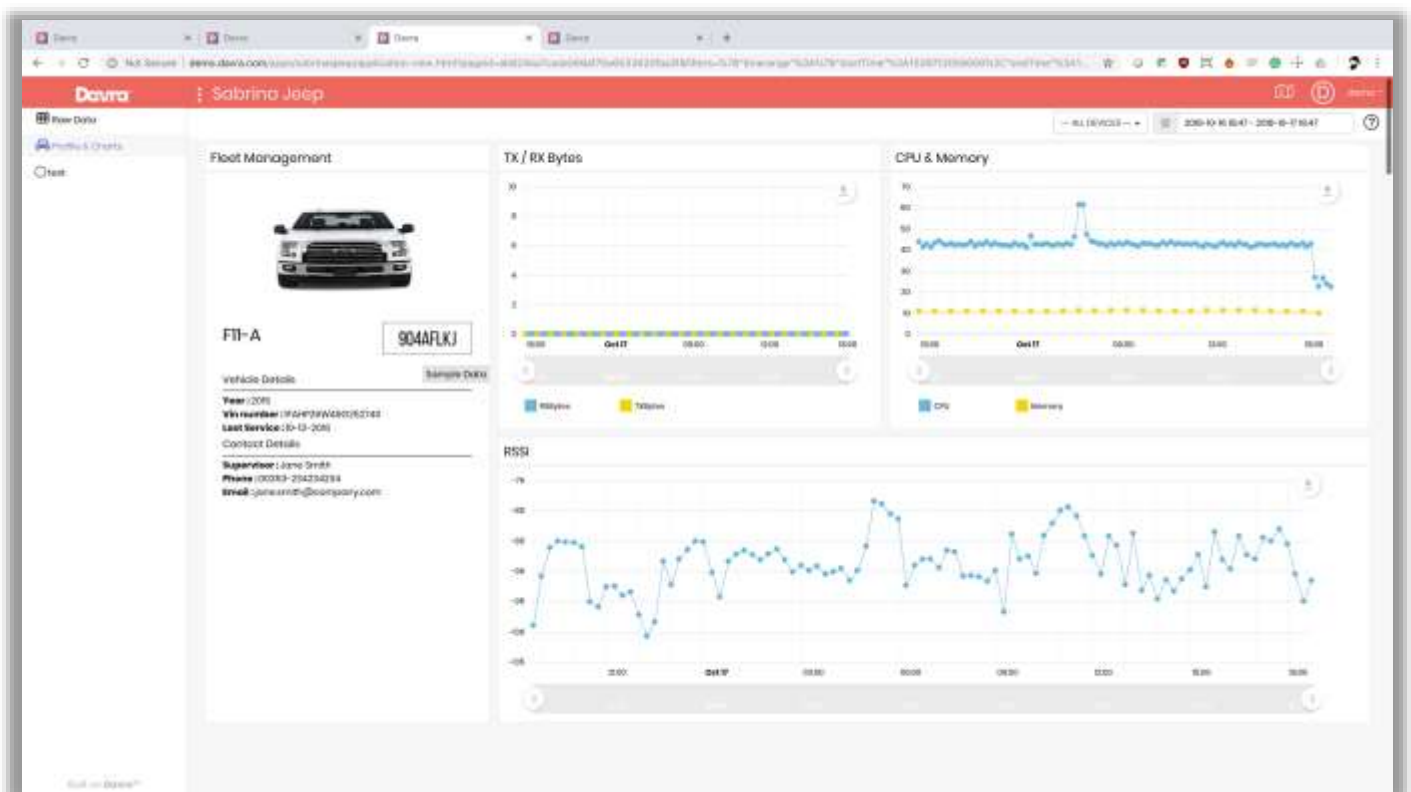
Clone Component

Frontend – Custom Widgets

If the gallery of built-in UI components is not sufficient for the project at hand, the Davra frontend development environment allows for new application development in HTML, CSS and JavaScript. Getting started is easy as existing UI applications can be cloned, amended and saved as new components.



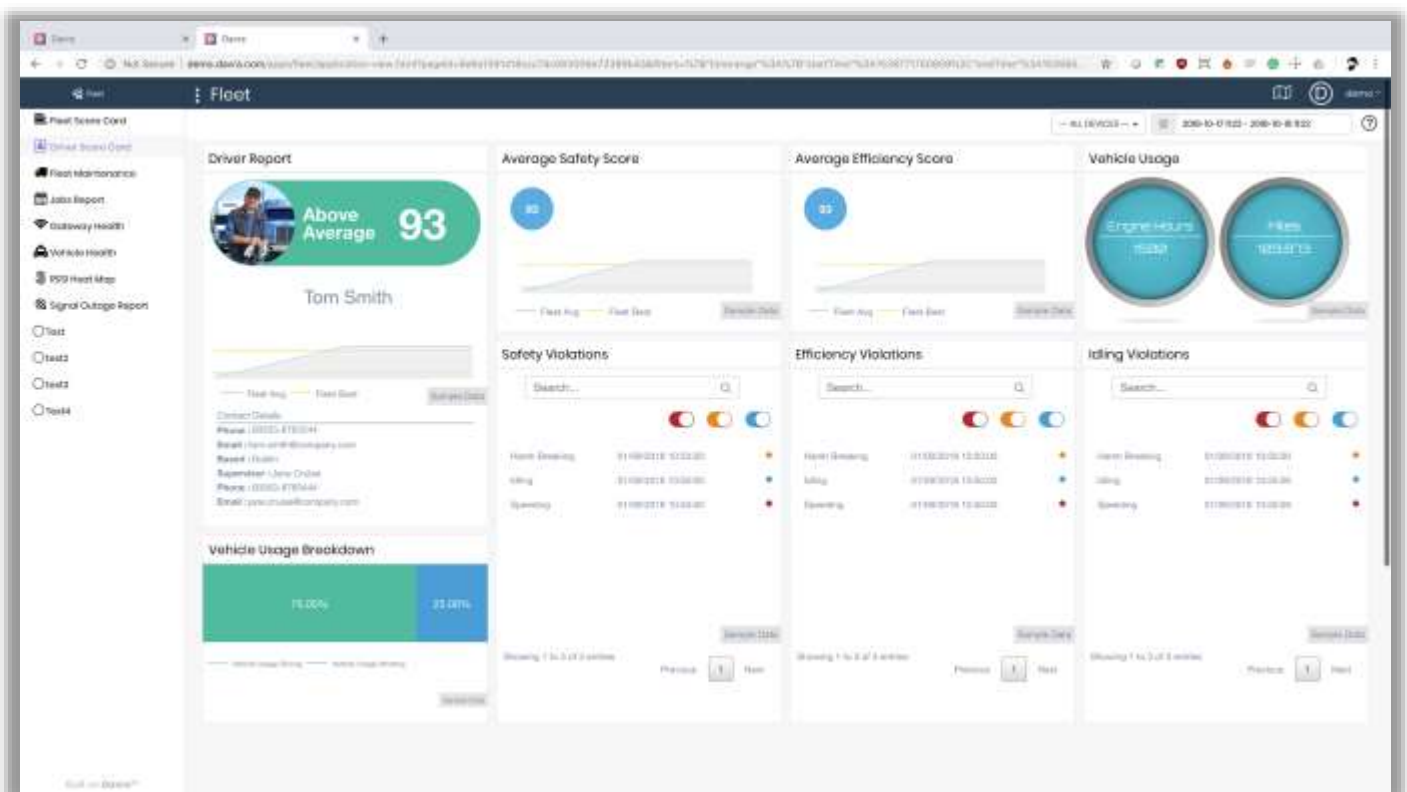
Editing Cloned Widget



Sample Custom Dashboard

6.4 Pages

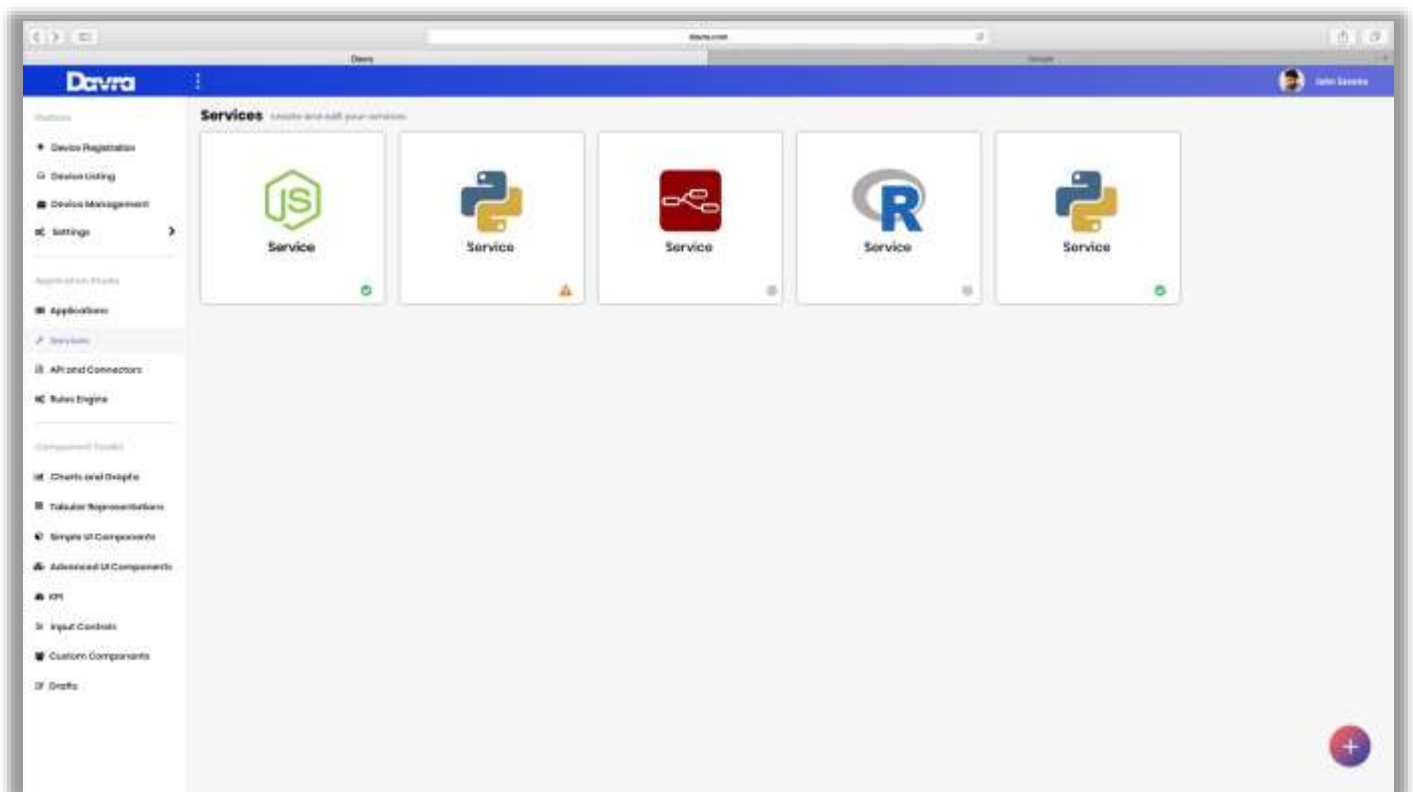
An application can contain multiple pages, as seen in the left-side menu below. Each page contains a number of widgets.



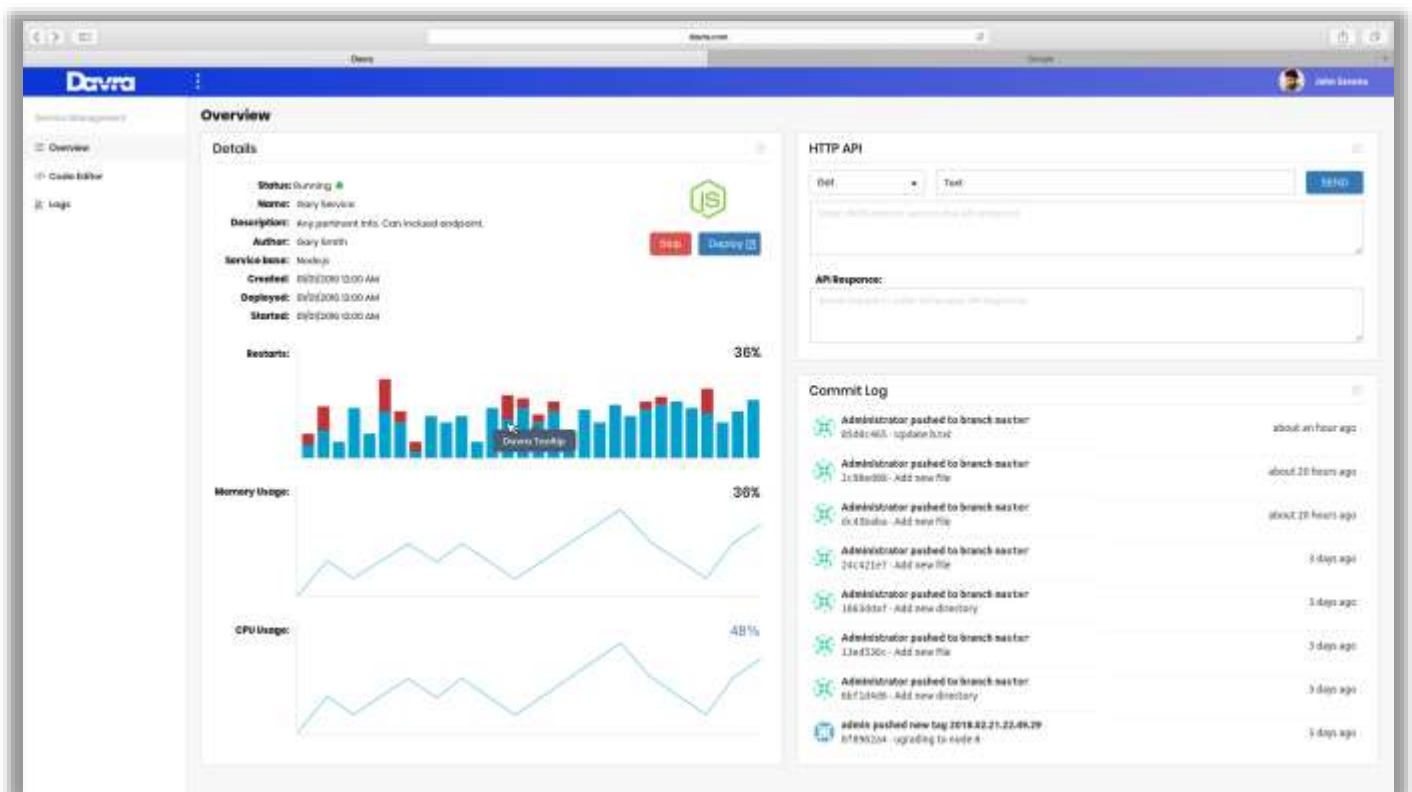
Example of pages within an Application

Backend Micro services

Davra allows for micro services development using Python, NodeJS or R. This is a powerful feature that gives technical customers free hands to build fully customised software on top of the platform, catering for most imaginable customer requirements.



Micro-Service Listings



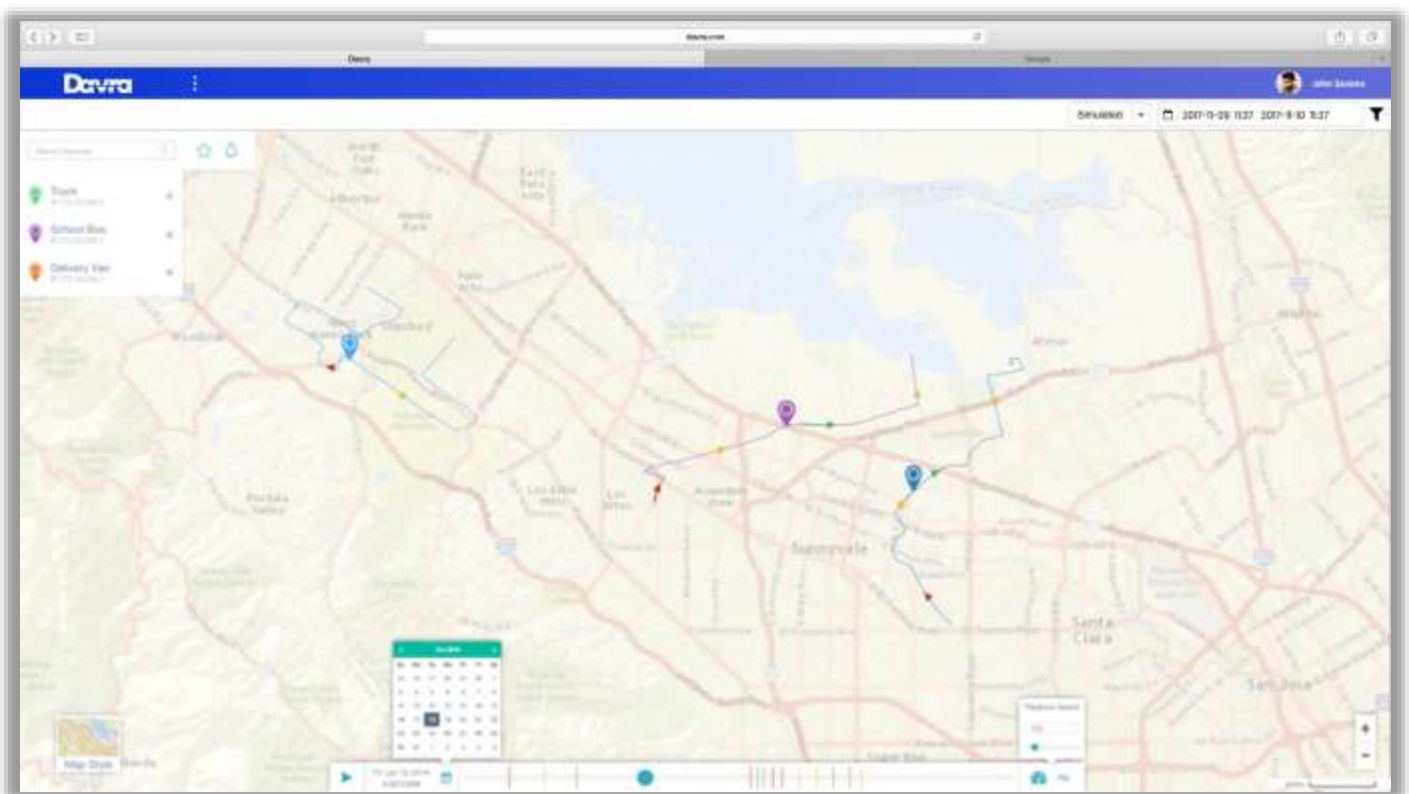
Micro-Service IDE

GIS – Geo-tracking and Geo-fencing

Geo-tracking and geo-fencing is available out-of-the-box with Davra, providing current location data, historical GPS replays and location-based alerting. GPS replays can include multiple devices simultaneously as seen below.

7.1 Geo-tracking

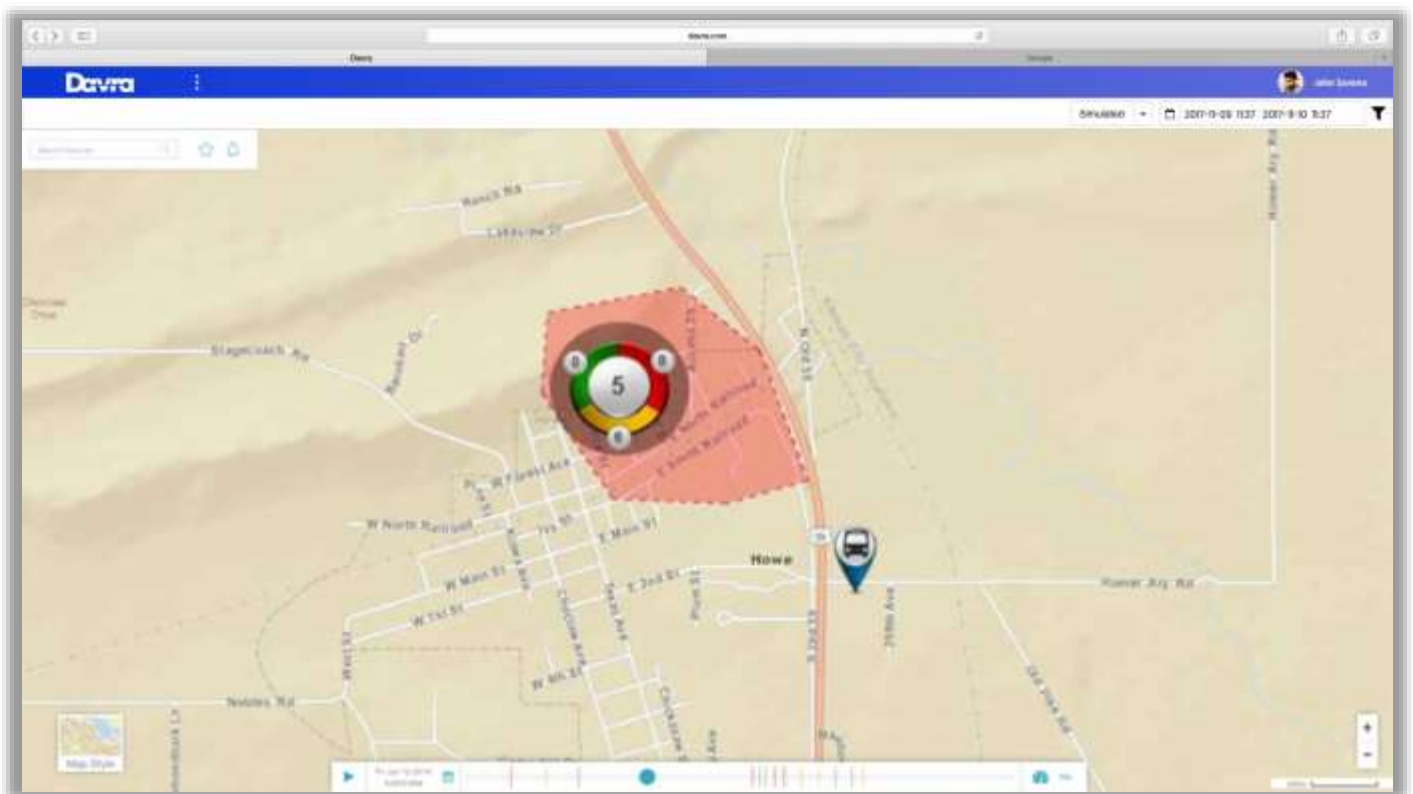
Once the device agent is installed and feeding location data to the ConnectThing platform it will appear automatically on the map and GPS replays containing historical location data will be available.



Geo-tracking 3 Assets

7.2 Geo-fencing

Geo-fences, as the one seen below, can be attached to specific devices and configured to generate for example email or SMS alerts.



Geo-fencing a cluster of Assets

Additional features

This document presents an overview of the primary features required to get started using the ConneCThing platform. To know more about additional features and capabilities in the platform, please visit the "Resources" section on www.davra.com or contact Davra at sales@davra.com for a demo session.