LoopBack

a highly-extensible, open-source Node.js framework
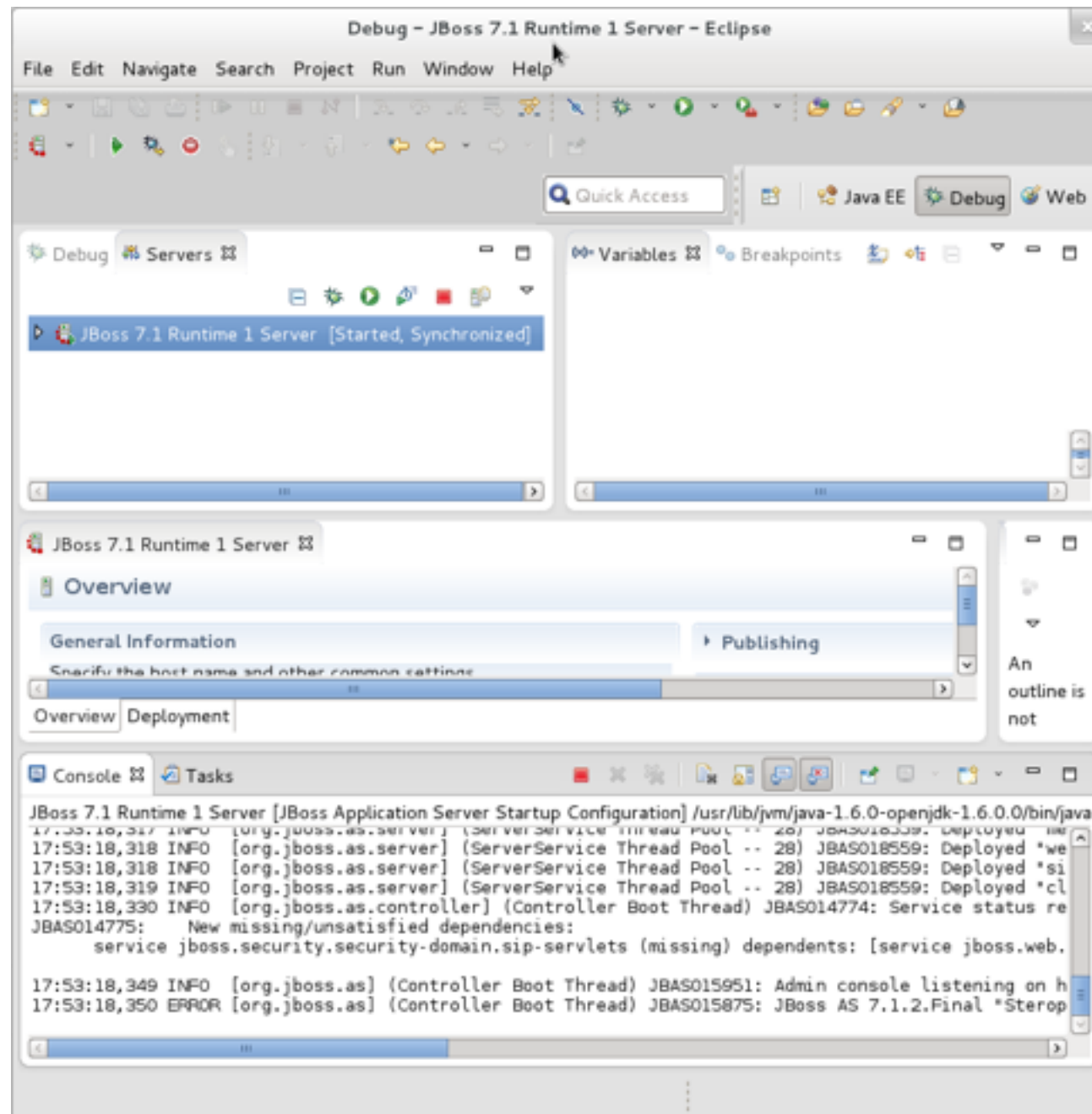
Eugene Terehov

Eugene Terehov

- **Co-Founder & CTO**
  bettervest.com (since 2013)

- **Founder und Orga**
  KarlsruheJS und FrankfurtJS

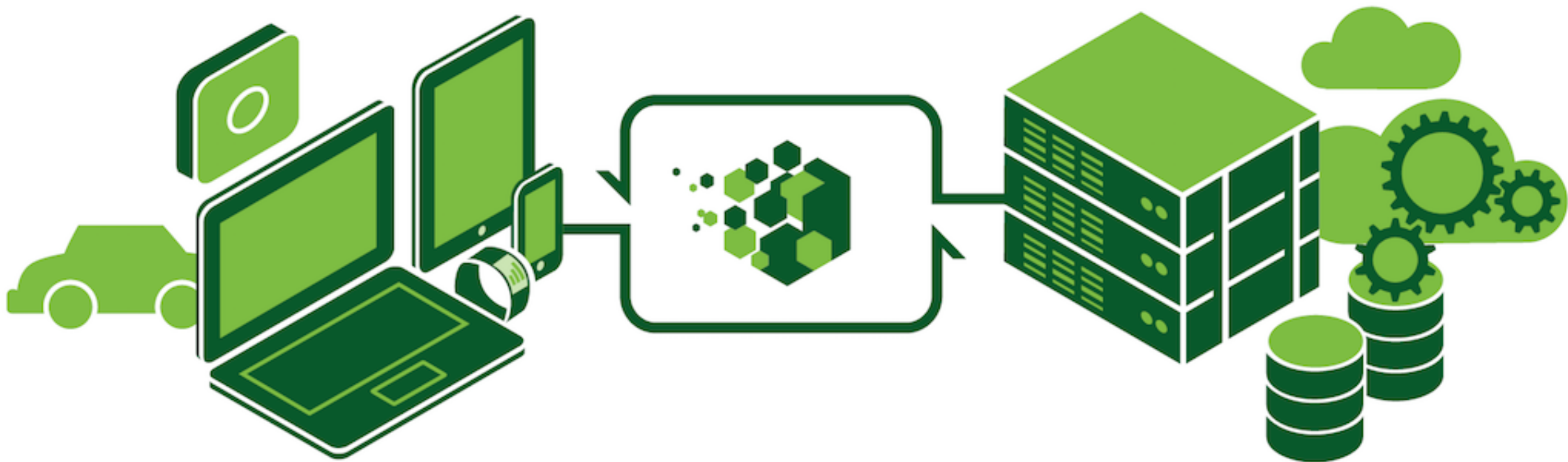- **Freelancer @ fullstack.build**

fullstack.build

build for web apps

vs.

lightweight & fast

divide and conquer

fullstack.build

Architecture

Channel SDKs

API Gateway

JSON API

LoopBack API Framework

PUSH · DEVICE · USER · REST API · FILE · OFF LINE · GEO

ORM Data Model

Enterprise Connectors

ORACLE

Microsoft SQL Server

mongoDB
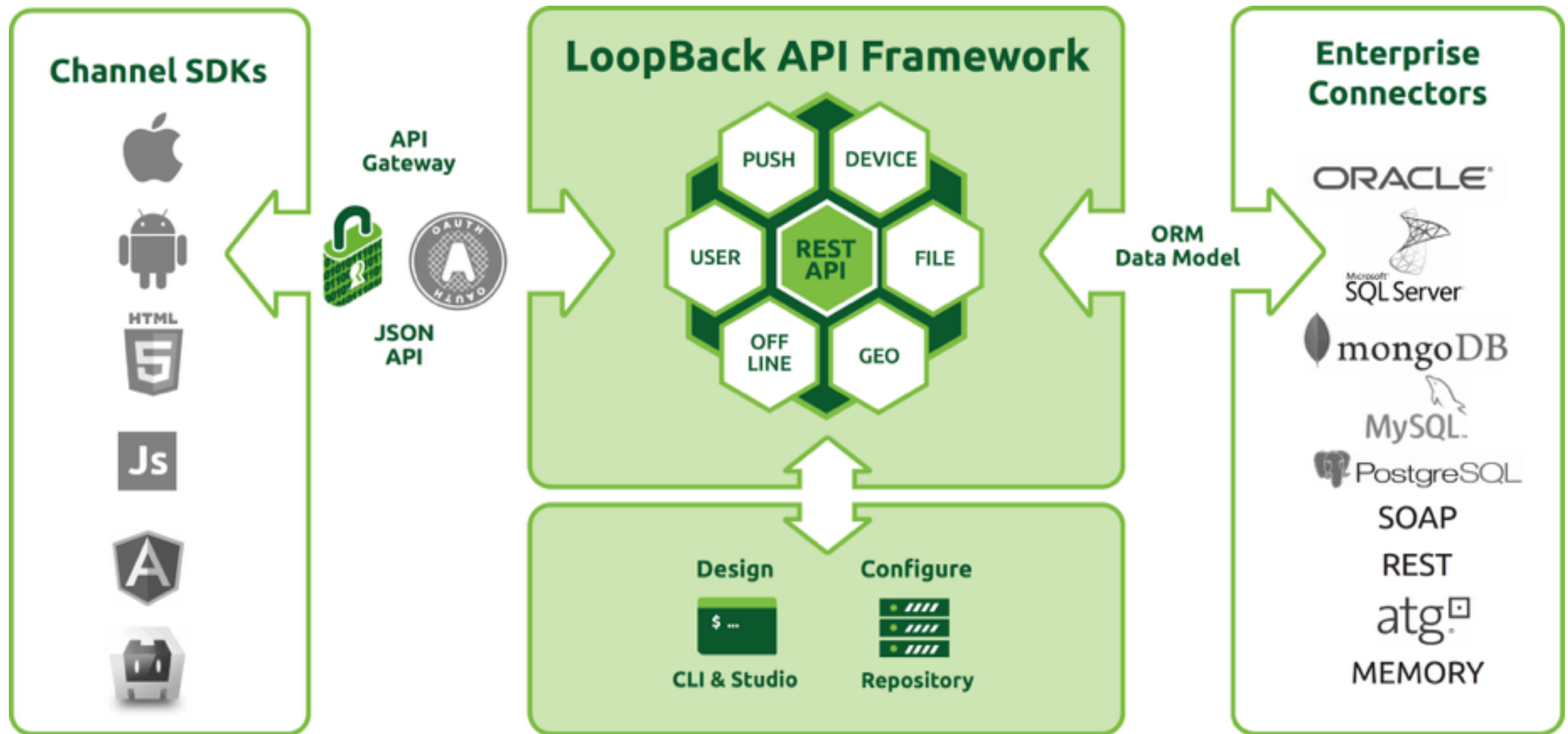
MySQL.

PostgreSQL

SOAP

REST

atg

MEMORY

Design
CLI & Studio

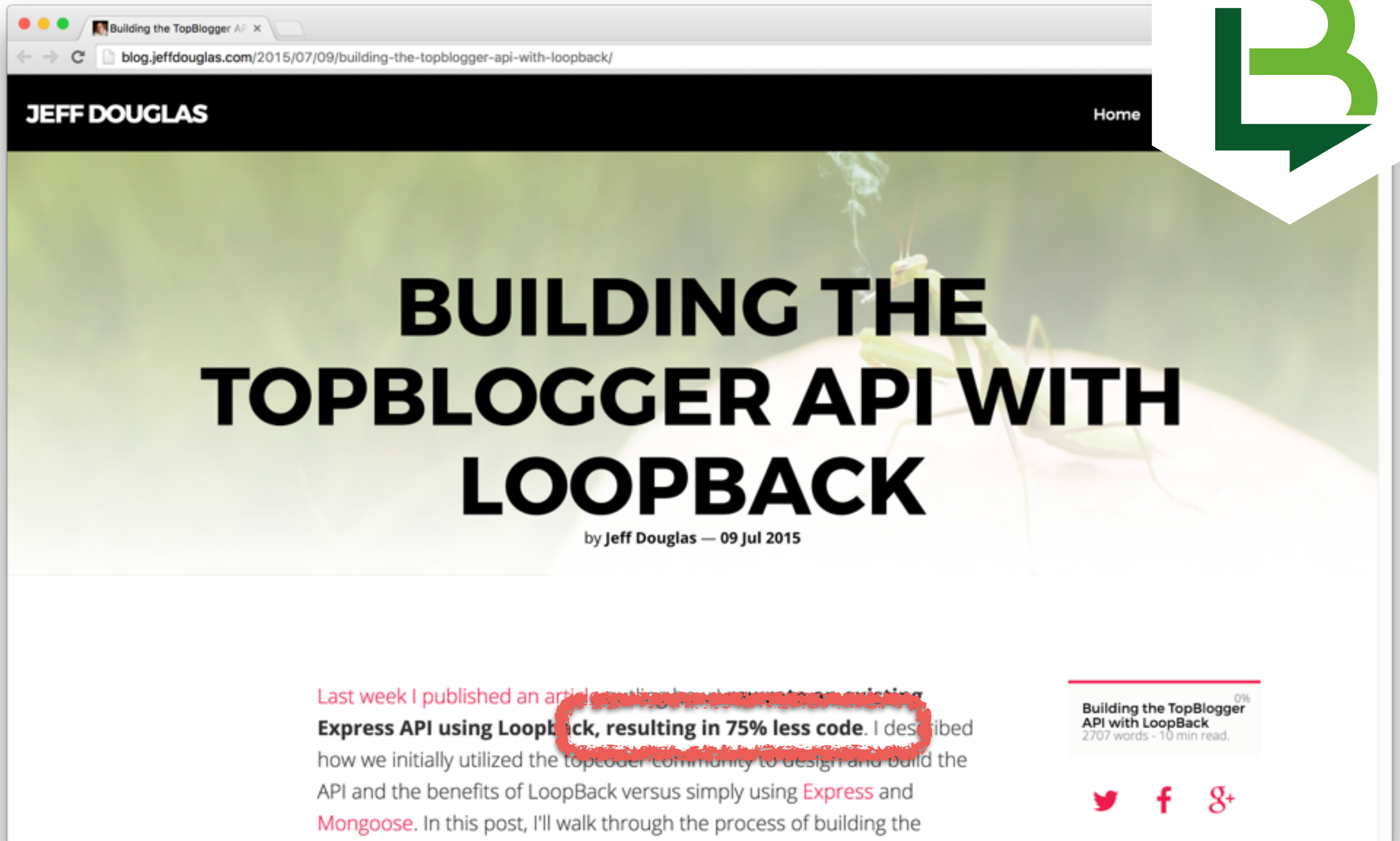Configure
Repository

fullstack.build

Overview

- Quickly create dynamic **end-to-end REST APIs**.

- On top of **Express**

- **Connect** devices and browsers to **data and service**.

- Built-in **API Explorer** for documentation and testing

- Built-in **role-based access controls** (ACLs)

- Use Android, iOS, and AngularJS **SDKs** to easily create client apps.

- **Add-on components** for push, file management, 3rd-party login, and geolocation.

- Use StrongLoop Arc to **visually edit, deploy, and monitor** LoopBack apps.

- **Professional support** by IBM StrongLoop (company behind Express)

fullstack**.**build

JEFF DOUGLAS

Home

BUILDING THE TOPBLOGGER API WITH LOOPBACK

by Jeff Douglas — 09 Jul 2015

Last week I published an article detailing how to migrate an existing Express API using Loopback, resulting in 75% less code. I described how we initially utilized the topcoder community to design and build the API and the benefits of LoopBack versus simply using Express and Mongoose. In this post, I'll walk through the process of building the

Building the TopBlogger
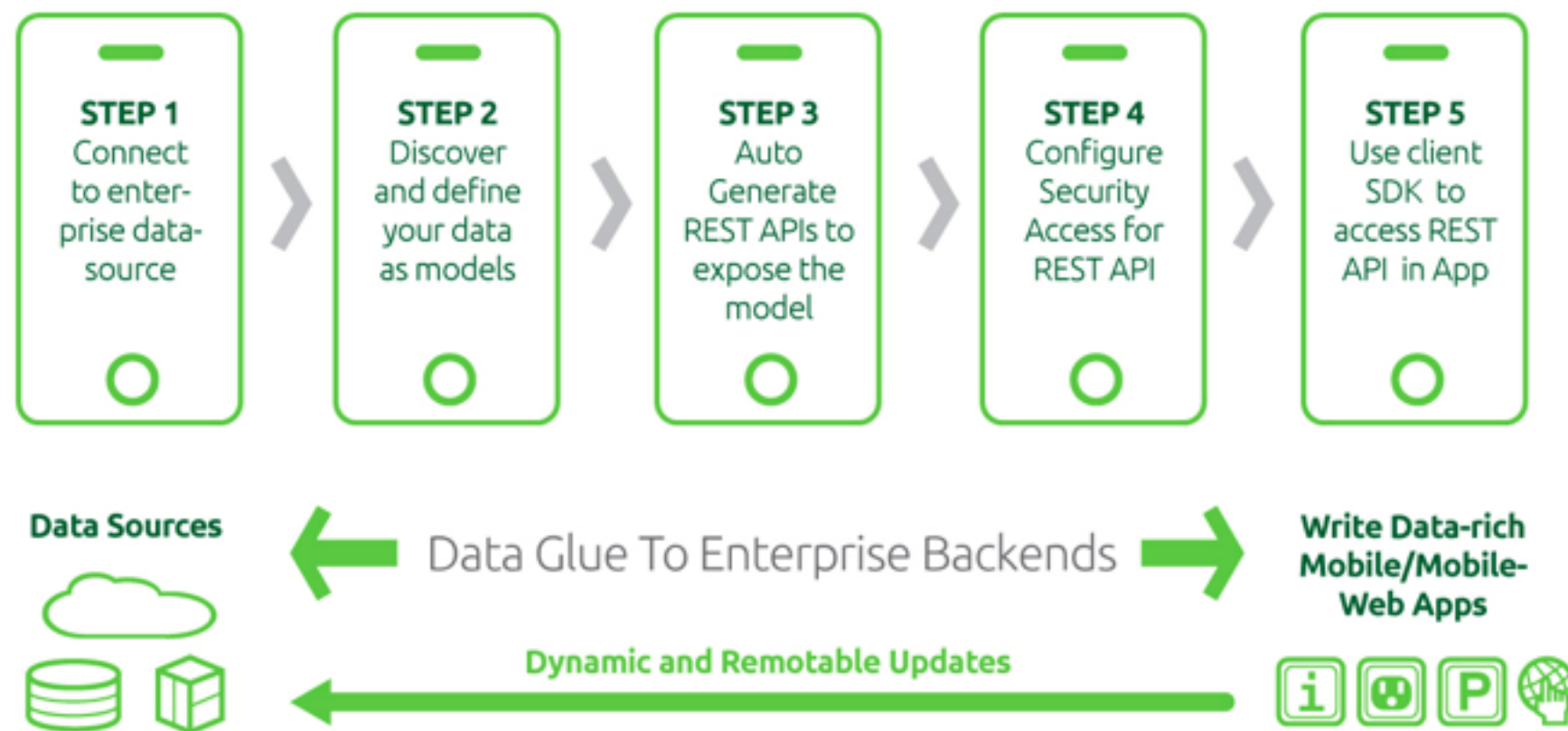API with LoopBack
2707 words · 10 min read.

fullstack.build

- Backend for mobile applications (native, web, and hybrid)

- Frontend for traditional enterprise systems

- Model = data + behavior.

- Isomorphic models: LoopBack, backend DBs, frontend

LoopBack: How It Works

STEP 1 Connect to enterprise datasource

STEP 2 Discover and define your data as models

STEP 3 Auto Generate REST APIs to expose the model

STEP 4 Configure Security Access for REST API

STEP 5 Use client SDK to access REST API in App

Data Sources

Data Glue To Enterprise Backends

Write Data-rich Mobile/Mobile-Web Apps

Dynamic and Remotable Updates

fullstack.build