Customise your low-code Platform

An Interactive Workshop on using MDE in the zAppDev low-code extensibility framework

What is zAppDev

- An MDE approach to low-code development advocating a generative approach to software development
- Three main components
  - MDE Visual Editors
  - zAppDev Metamodel (Application specification/semantics model)
  - Code Generators (called: Implementation Strategies)

zAppDev

MDENet | The home of model-driven engineering

High Level Metamodel

DATA	INTEGRATION	BUSINESS	SERVICE	UI/UX
ENTITIES	MESSAGING	WORKFLOWS	REST API	FORMS
	DATABASES	OPERATIONS	MANAGEMENT	THEMES
	REST/SOAP		DATA CONTRACTS	

zAppDev

MDENet | The home of model-driven engineering
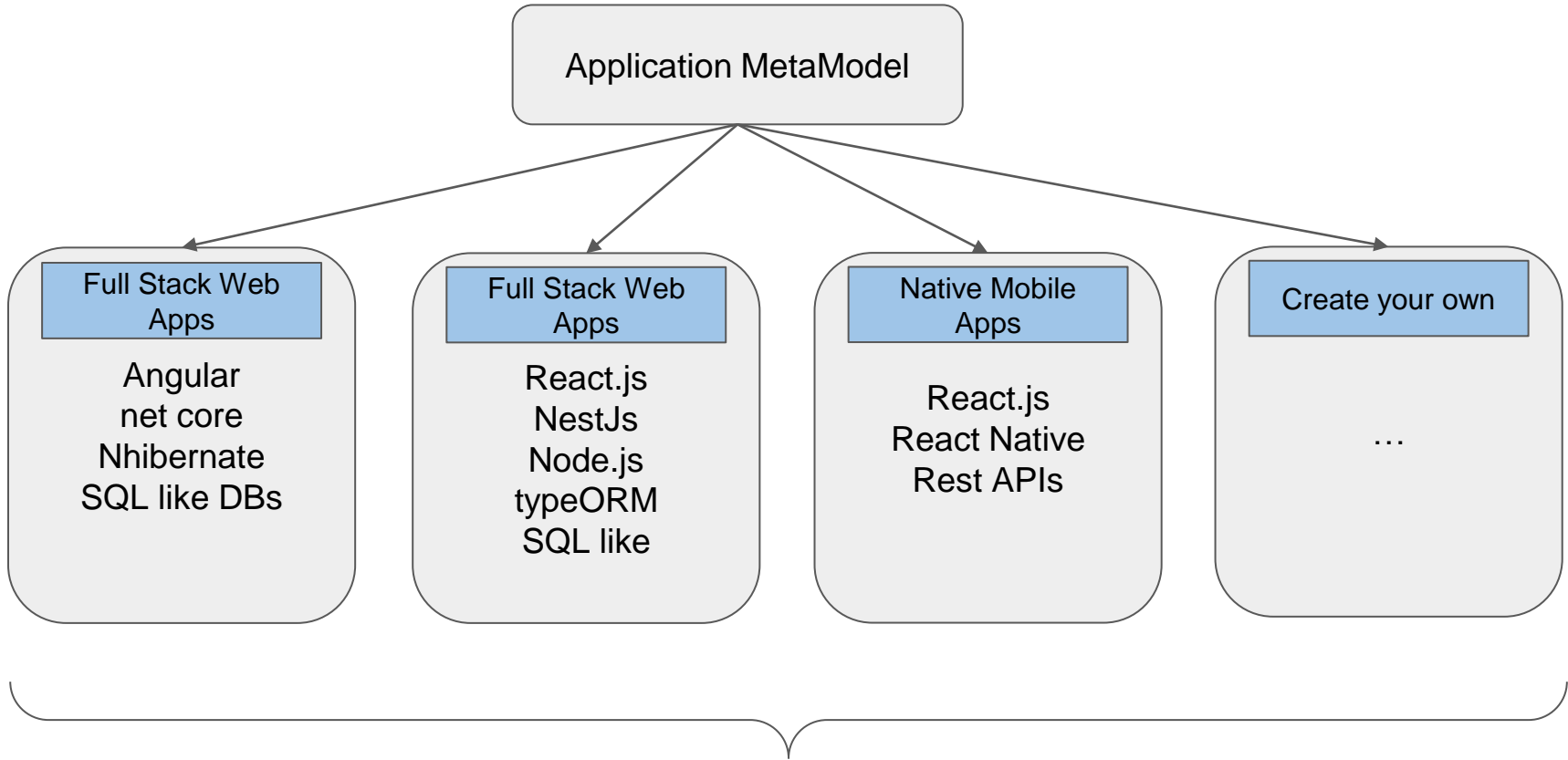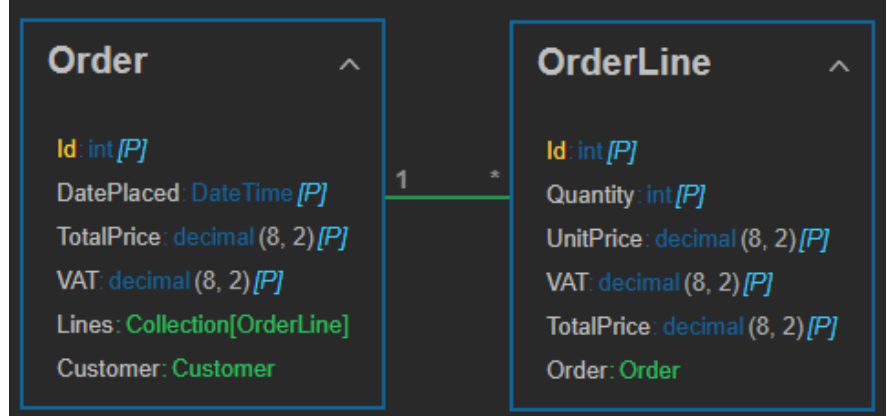
Implementation Strategies

- Completely separated from the metamodel
- No assumption of any runtime libraries
- Based on open source technologies/frameworks
- Extensible
- Relatively easy to modify
- Can be written in any language
- No risk of vendor lock-in:
  - Not tied into a specific platform infrastructure and technology

Application MetaModel

Full Stack Web Apps

Angular
net core
Nhibernate
SQL like DBs

Full Stack Web Apps

React.js
NestJs
Node.js
typeORM
SQL like

Native Mobile Apps

React.js
React Native
Rest APIs

Create your own

…

Different Implementation Strategies

zAppDev

MDENet | The home of model-driven engineering

Models we will generate

- Four classes
  - Customer
  - Address
  - Order
  - Order Details

- Service layer with CRUD Customer operations exposed as a REST service

Existing Strategy Code Architecture

- We will generate code for the [NestJS](#) framework
- A progressive Node.js framework for building efficient, reliable and scalable server-side applications.
- We will focus on
  - Entities
  - Database persistence (using TypeORM)
  - Service layer
  - API Layer (REST)

zAppDev

MDENet | The home of model-driven engineering

Live Workshop - Extend Code Generation

- Generate code for the Data-contracts* zAppDev's feature
- Generate a class for each data contract with:
  - The properties user has selected
  - A function to convert an Entity to a DTO
  - A function to convert a DTO to an entity
- Alter the API to convert Entities to DTOs using these functions

Data-contracts are visually defined DTOs that specify which properties
of the domain model will be exposed through the API Layer

Preparation

- git clone git@github.com:zappdev/zAppDev.Strategy.nestjs.git
- Open solution with Visual studio or Visual Studio code
- Dependencies: .NET 6, node.js (>= 14), mysql or mariaDB
- Create an empty database

More about us…

- Find more about zAppDev at zappdev.io

- You can create a free account at ide.zappdev.io

- Feel free to contact us at support@zappdev.com

Thank you!

zAppDev Team

Dimitris Moudilos
Software Engineer

Ilias Georgakis
Software Engineer