CIKLUM | SPEAKER'S CORNER

OCTOBER, 11   17:30 (CET)   ZOOM

ENHANCING NODE.JS ARCHITECTURE:
EXPLORING NESTJS

GORGAN DANIEL-MIHAI

SENIOR JAVASCRIPT
DEVELOPER AT CIKLUM

Meet Ciklum

CIKLUM

We empower companies to meet their digital initiatives by providing end-to-end software, integration and innovation services

Our services:

Digital Commerce

Intelligent Automation

ITO & Managed Service

Data & Analytics

Cloud

Engineering Services

2002 founded

4000+ professionals

20+ offices

300+ clients

Leading companies choose us:

JUST EAT Takeaway.com   METRO MARKETS   eToro   Z ZURICH   pay   Greensill   KANTAR RETAIL   dacadoo

Speaker:

Daniel Gorgan

Senior JavaScript Developer at Ciklum

- Tech enthusiast specializing in Node.js, SQL/NoSQL and Cloud technologies with 5+ years of experience
- Hands-on experience in projects across outsourcing and product companies contributing to the development of in-house products, smart chatbots and voicebots by leveraging different AI technologies

CIKLUM

Agenda

01 | Code architectural challenges
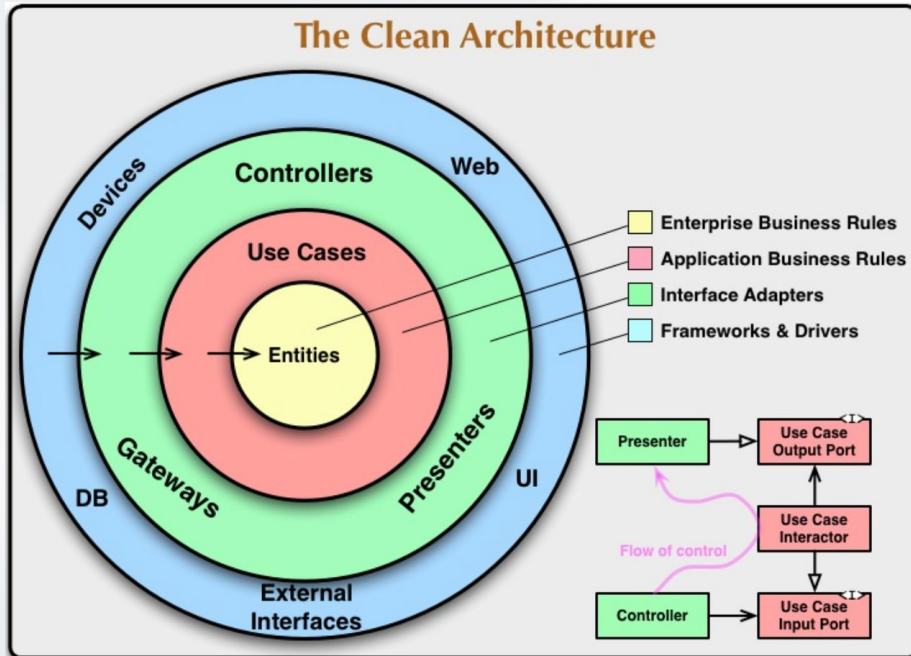
02 | NestJS

03 | Demo

04 | Conclusions

The most common architectural problems

Found architectural problems

- No separation of concerns
- Hard to maintain the code
- Hard to test the code
- Adding external dependencies interfere with the business logic code
- No standards and consistency to follow

Clean Architecture

Node Clean Architecture Diagram, Image by Robert C. Martin

NestJS nest

More about

Released in 2016

A Node.js framework for building efficient, scalable and maintainable server-side applications

It combines elements of:

- OOP (Object Oriented Programming)
- FP (Functional Programming)
- FRP (Functional Reactive Programming).

Nestjs is not Nextjs

nest vs **NEXT.JS**

API development   Server Side Rendering

Nestjs is not Express.js

nest vs express

Opinionated   Unopinionated

nest

Comparison

https://npmtrends.com/@hapi/hapi-vs-@nestjs/core-vs-express-vs-fastify-vs-koa-vs-sails

Key features

- Modular architecture inspired by **Angular**
  - Decorators
  - Dependency injection

- **Typescript** integration
  - JS support as well

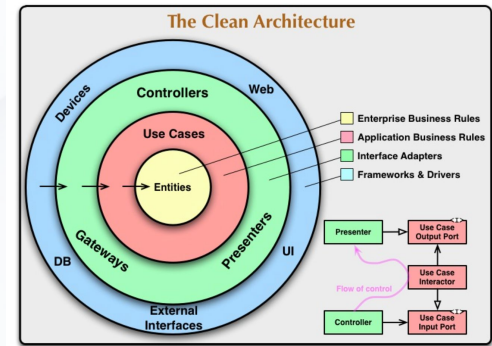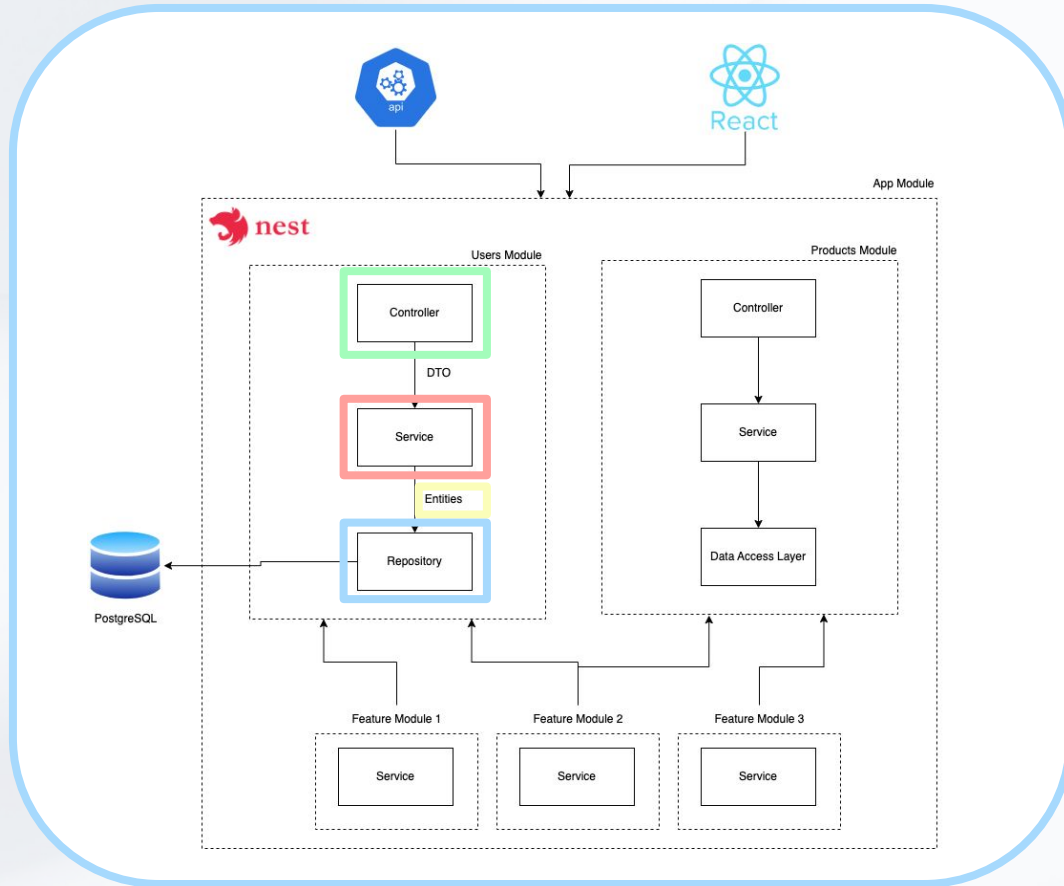# Key features

Extensive ecosystem and built-in modules

- Nest CLI (optional: monorepos)
- REST, gRPC, GraphQL
- MVC
- WebSockets
- Microservices
- RabbitMQ / Kafka
- Authentication and authorization
- Different databases support (sequelize, TypeORM, mongo)
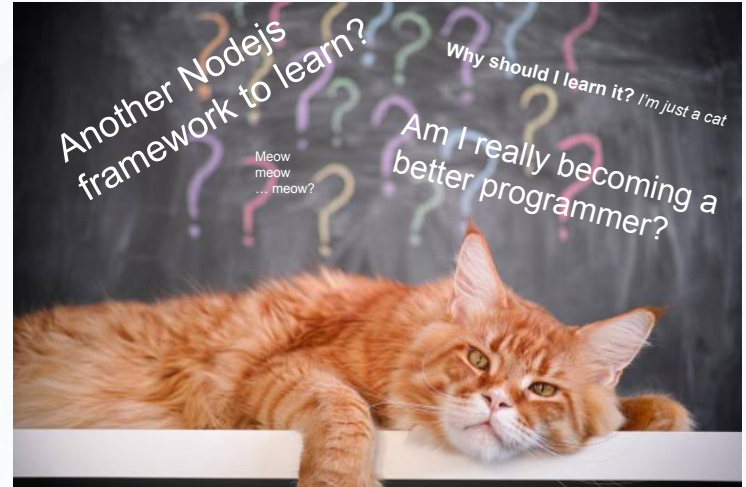- Testing utilities and best practices

Simplified (initial) Nest architecture
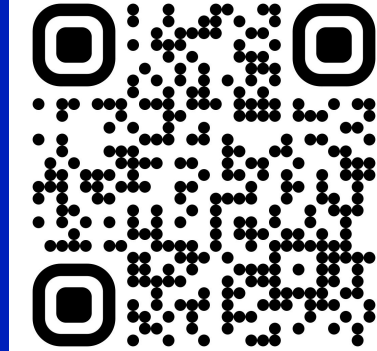
Demo 💻

Is it worth Learning Nestjs?

What industry thinks about it

- 3-5 years of relevant work experience with at least 2 years of proven experience in BackEnd Development of Web and Mobile applications with NodeJS and Nest

  - NodeJS microservices with Typescript and NestJS

- Back-end Node.js, NestJS, Python

  - Knowledge of **Node.js** and frameworks available for it (**Nest.js**)

- Demonstrated experience with back-end development using NestJS.

Any questions?

Share your feedback!

Thank you!

Gorgan Daniel
dagor@ciklum.com | www.ciklum.com