Using Hasura
to speed up development
in a FinTech startup

Hello!

Pamela Gotti

Software engineer

- Senior software engineer in Credimi
- CTO of She Tech Italy
- Lindy hopper and jazz lover

Credimi

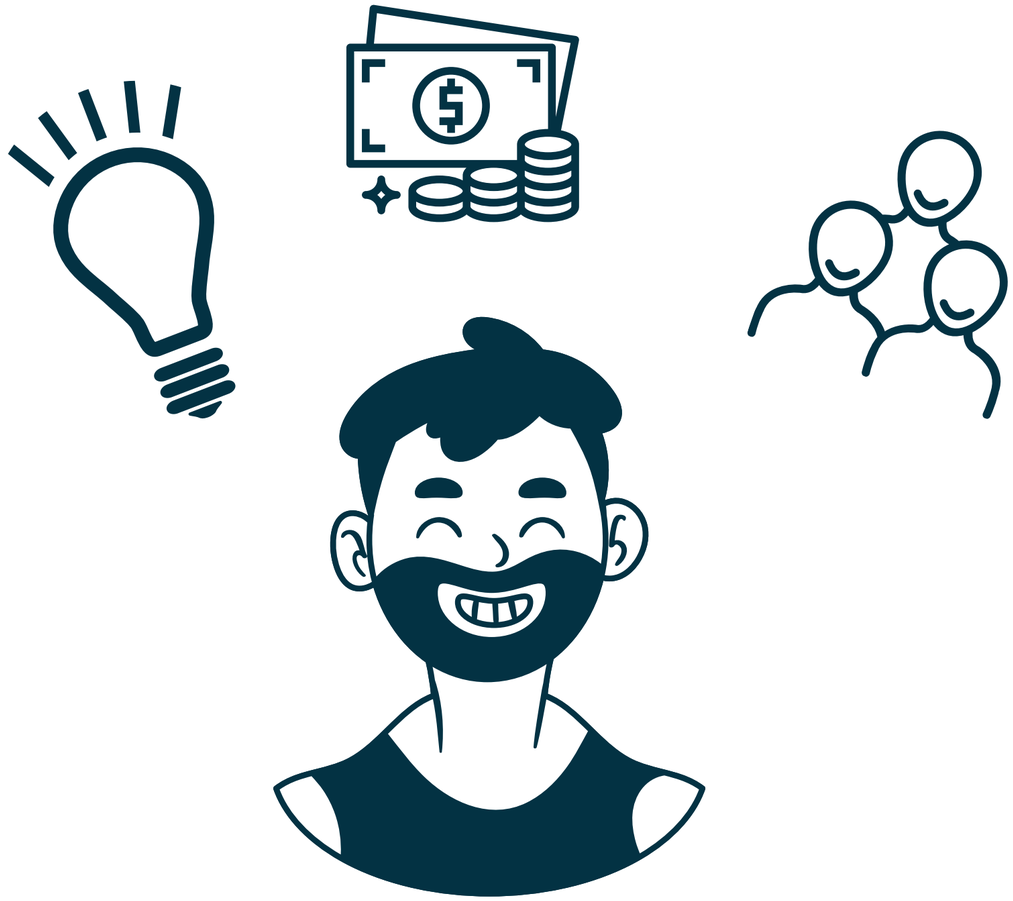
- Italian startup based in Milan born in 2016
- Leader digital lender in continental Europe
- 928M € loans
- 41k applications

Highlights

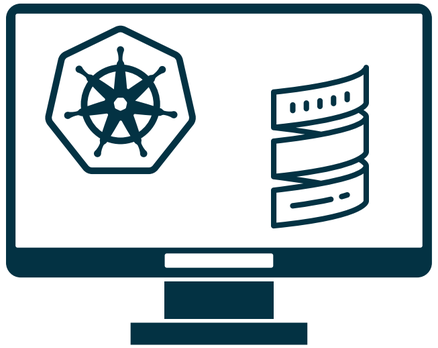
Time to market for new features decreases

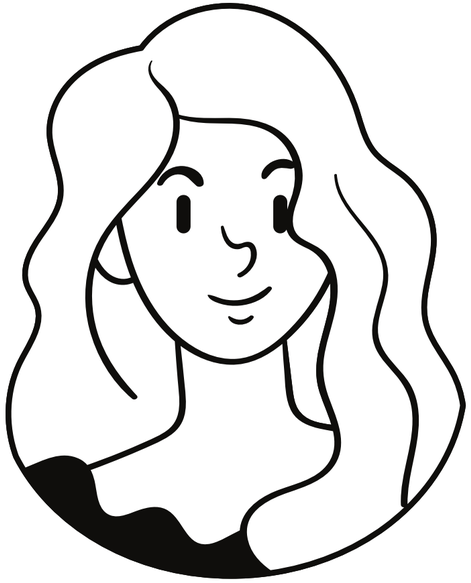
Developers focus on delivering business value

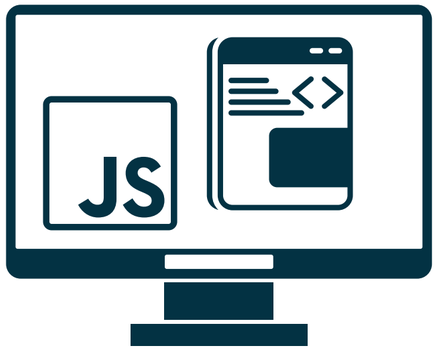
Once upon a time...

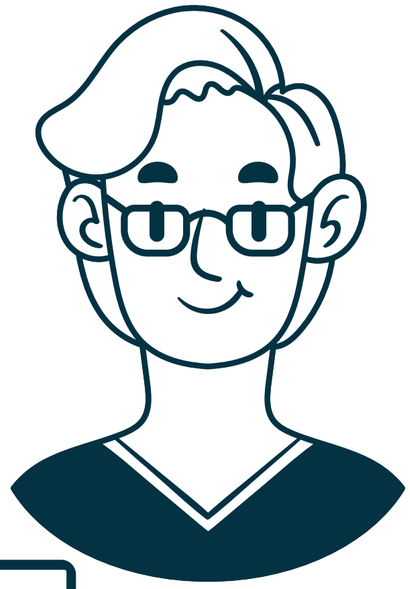


Once upon a time...

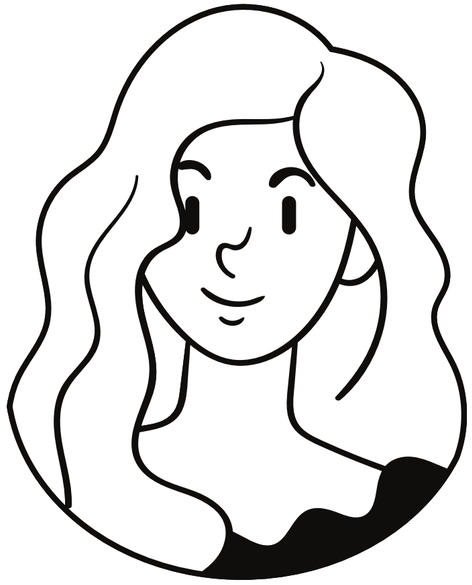


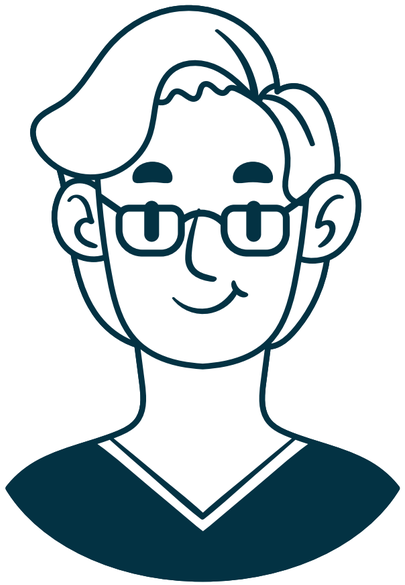




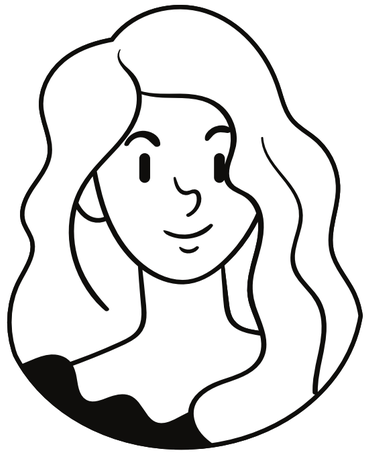


Once upon a time...





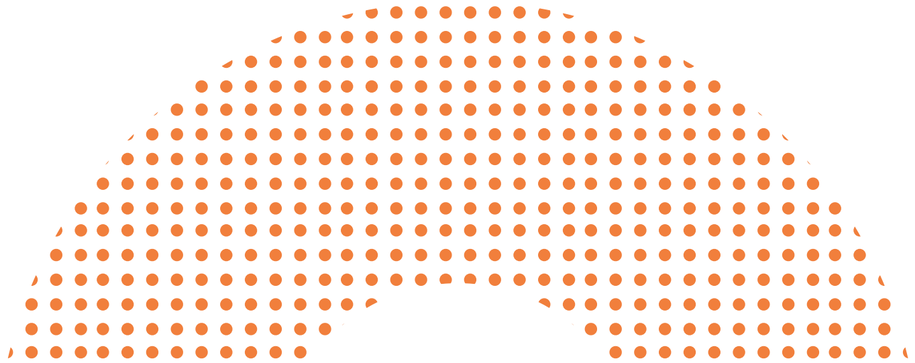
Once upon a time...



Should we run away as far as possible?

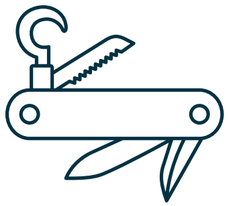
Definitively

Developers Needs



Go fast

Quick release cycles to bring features to our customers



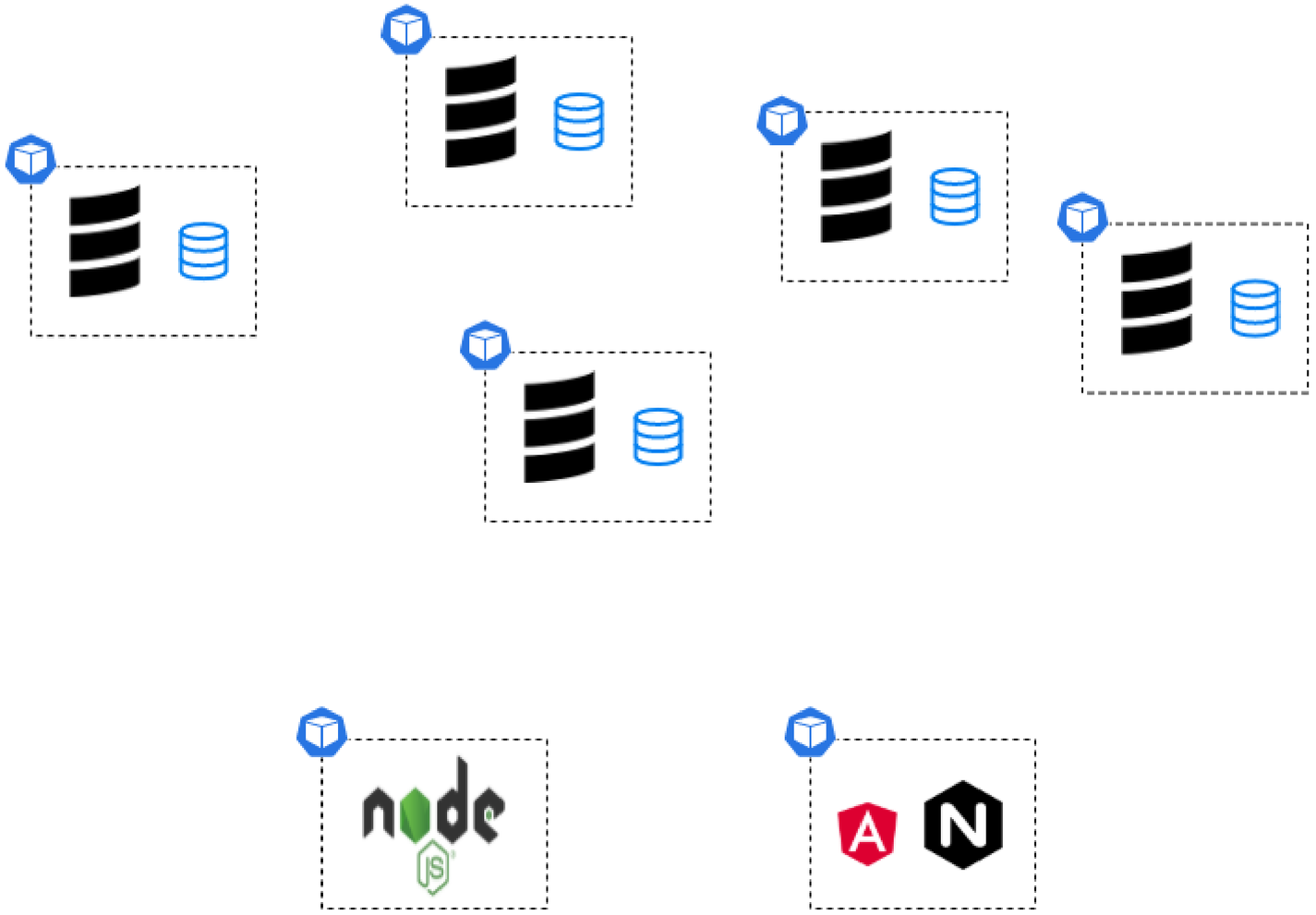
Versatility

Bringing a new feature live should be easy

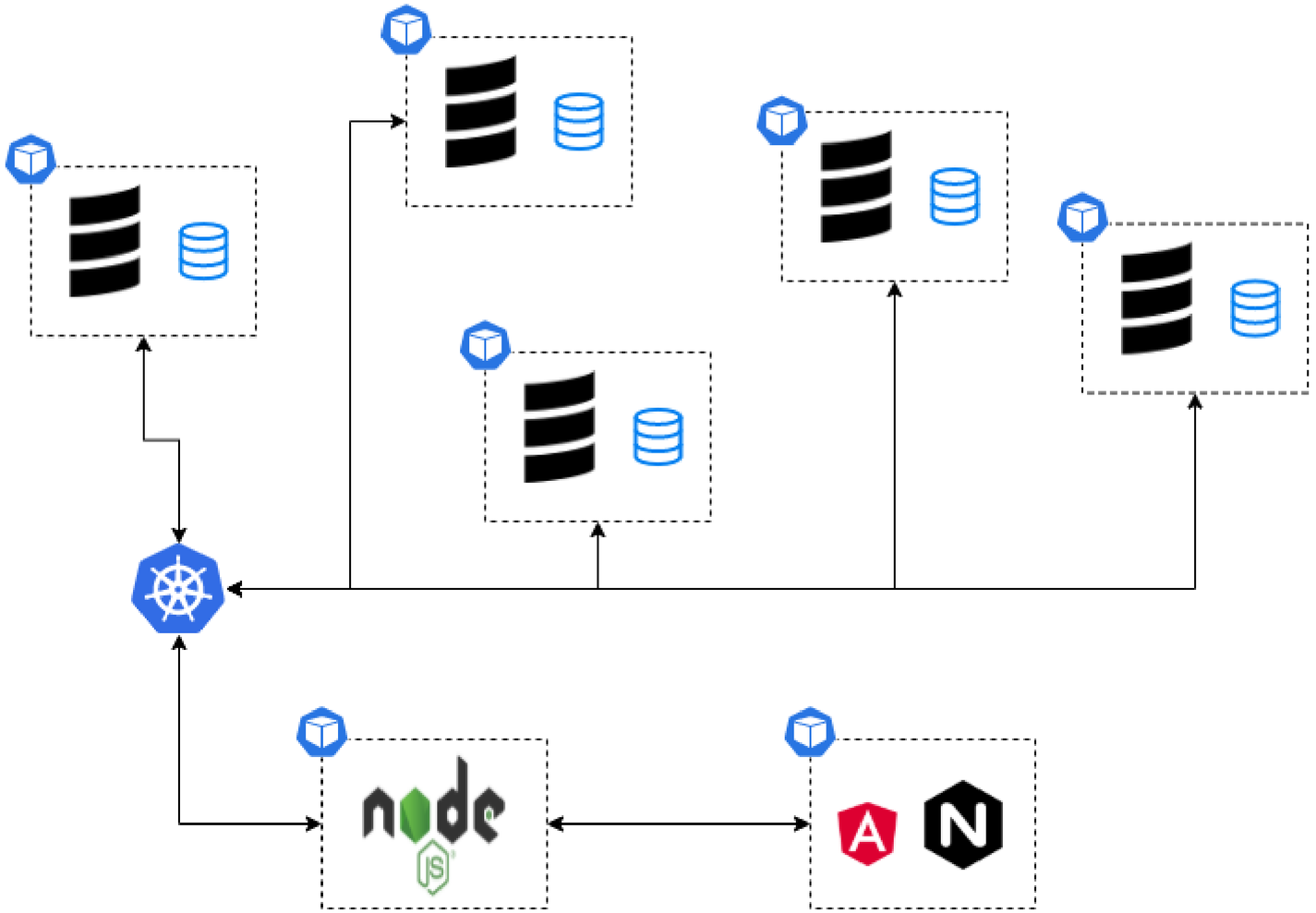
Focus on value

Developers should focus on core features, not boilerplate

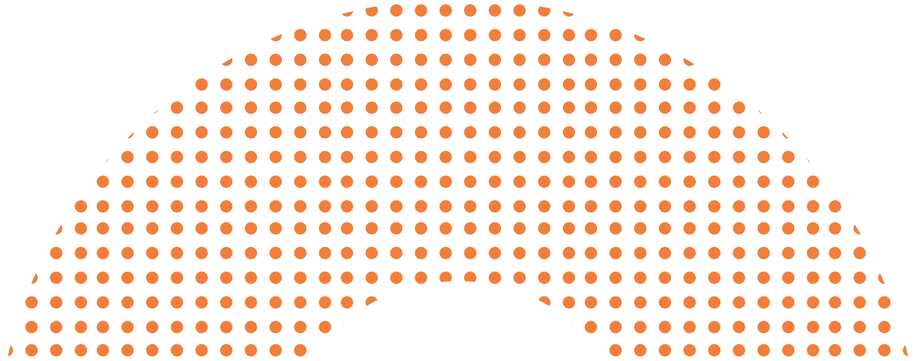
Credimi 1.0



Credimi 1.0

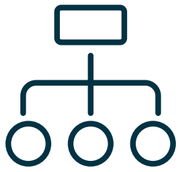


Problems with Credimi 1.0



Boilerplate

At API and Nginx layer



No schema

No easy way for frontend
and backend to share schema

Authorization

Field level authorization not easy
to achieve

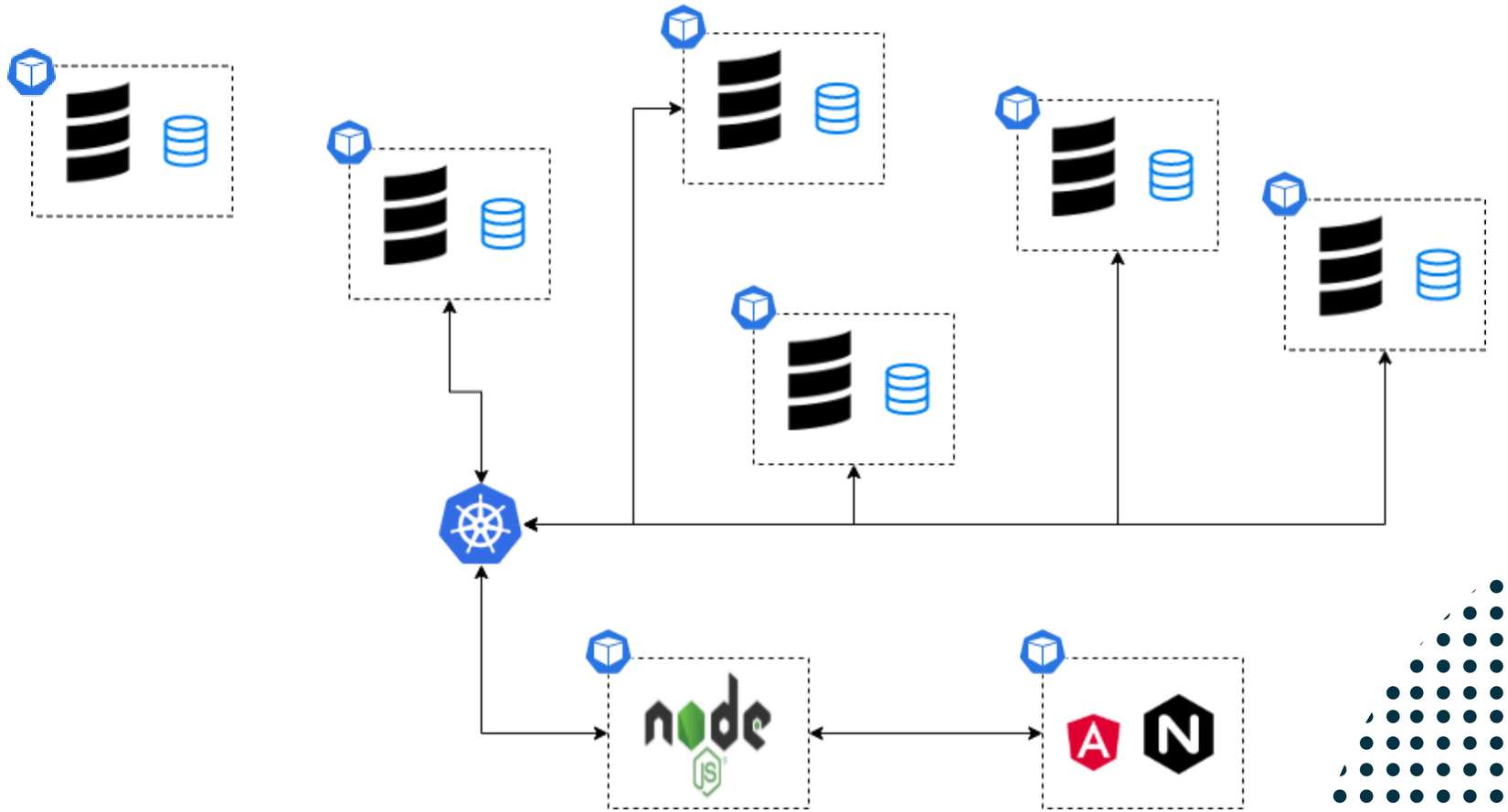
Heavy APIs

Representation of some resources
required heavy computations
on the backend

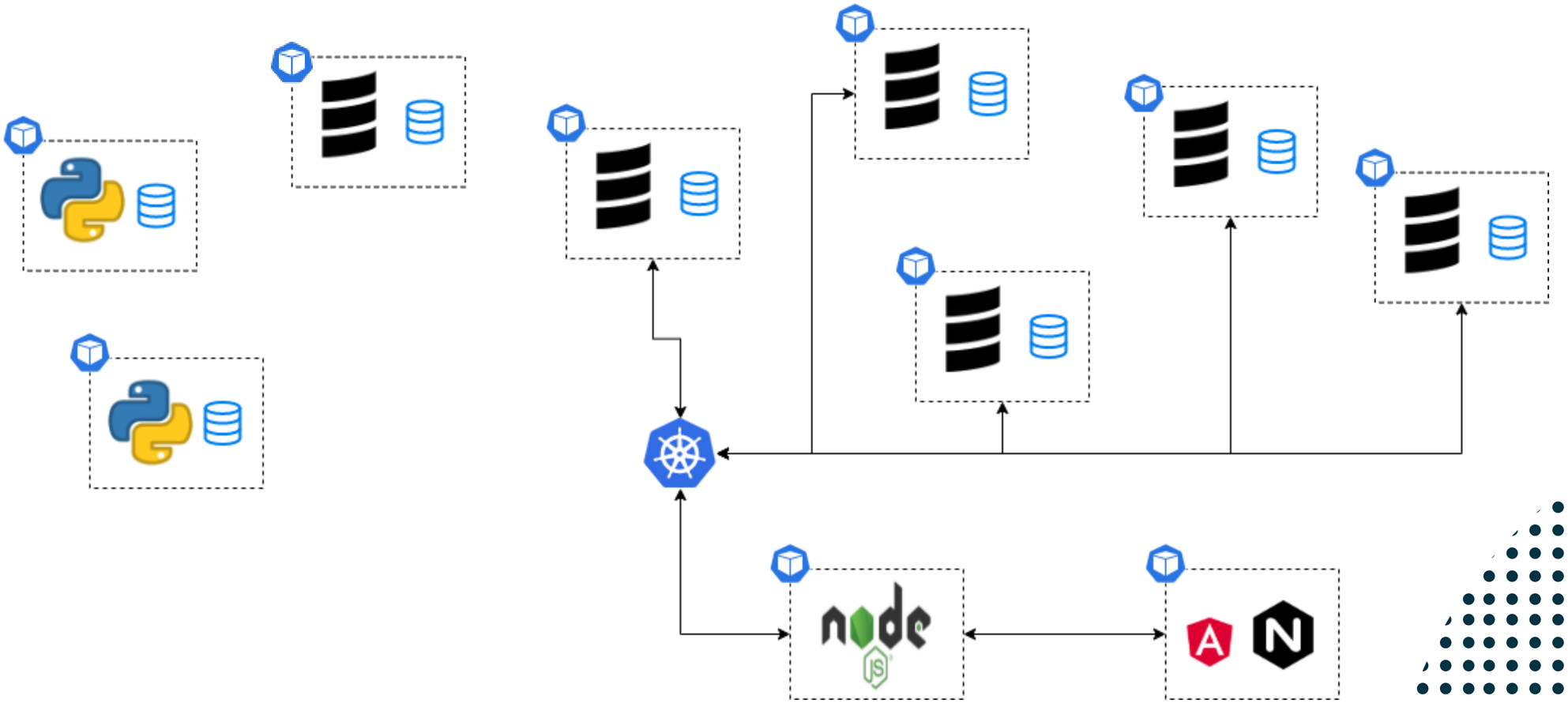
In the meanwile...

Credimi was in continue evolution

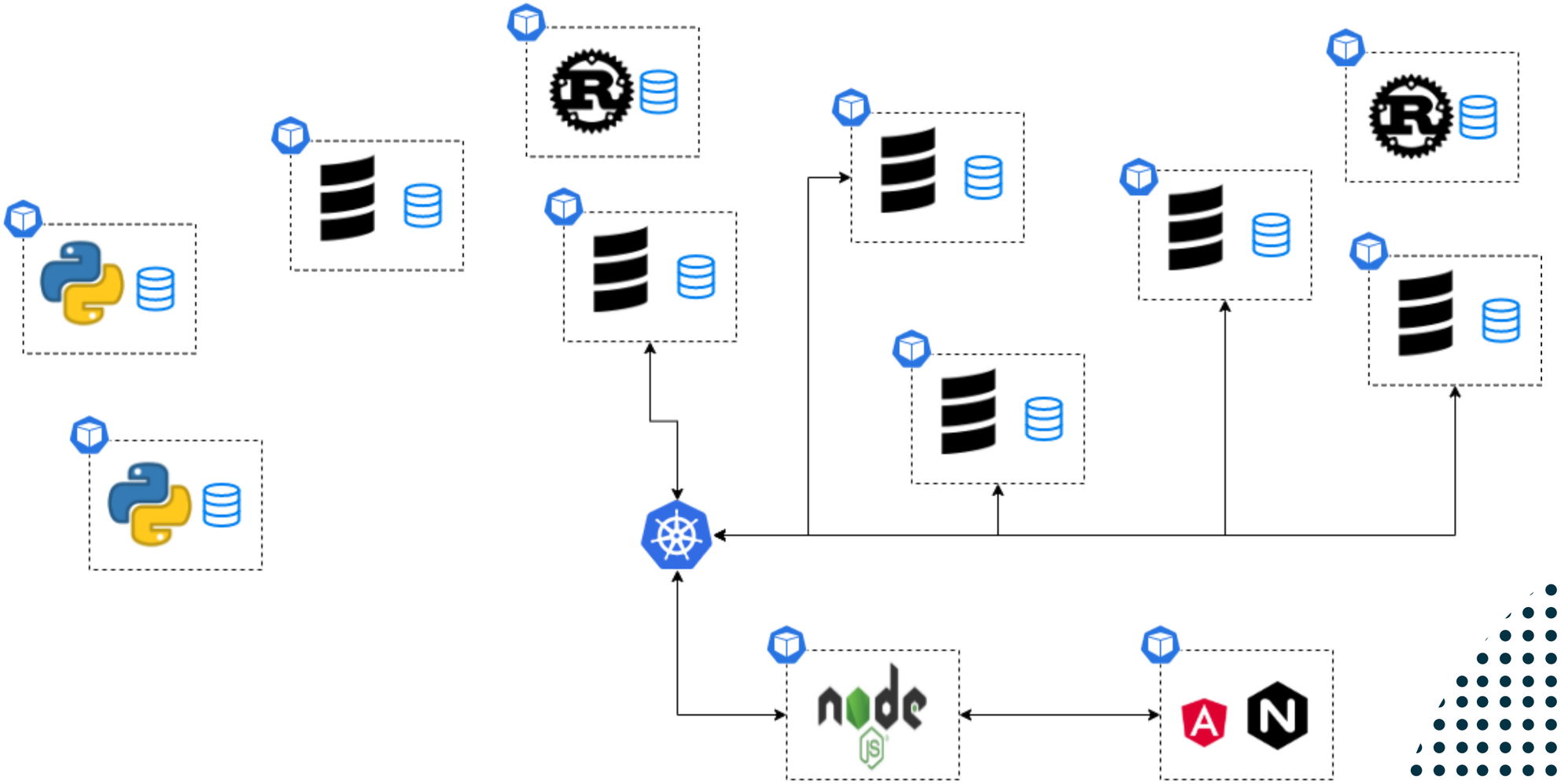
Credimi 1.1



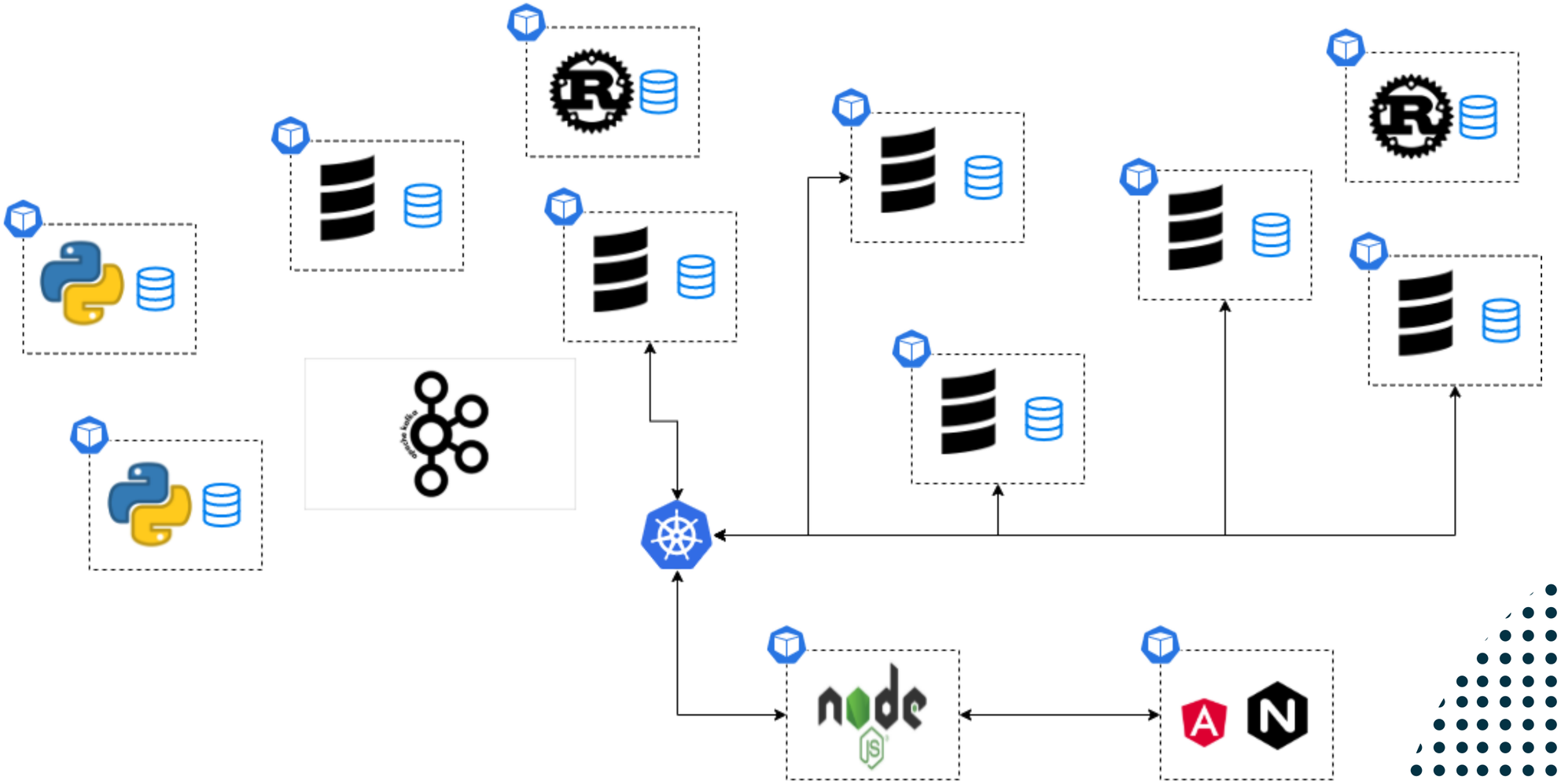
Credimi 1.1



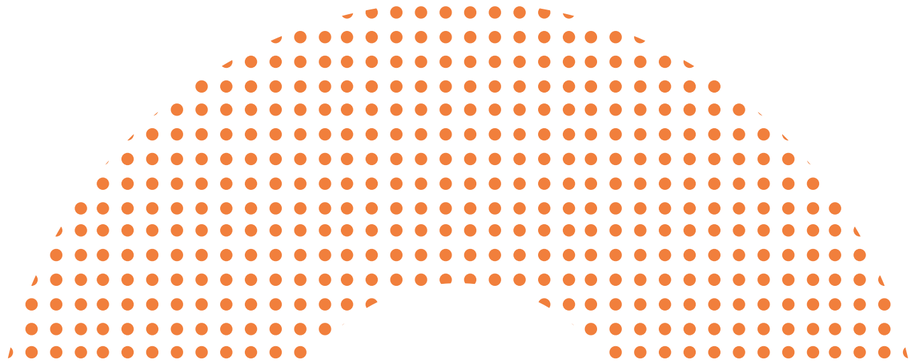
Credimi 1.1



Credimi 2.0



Opportunity



Event sourcing

Exploit Kafka as event collector to create a read side database

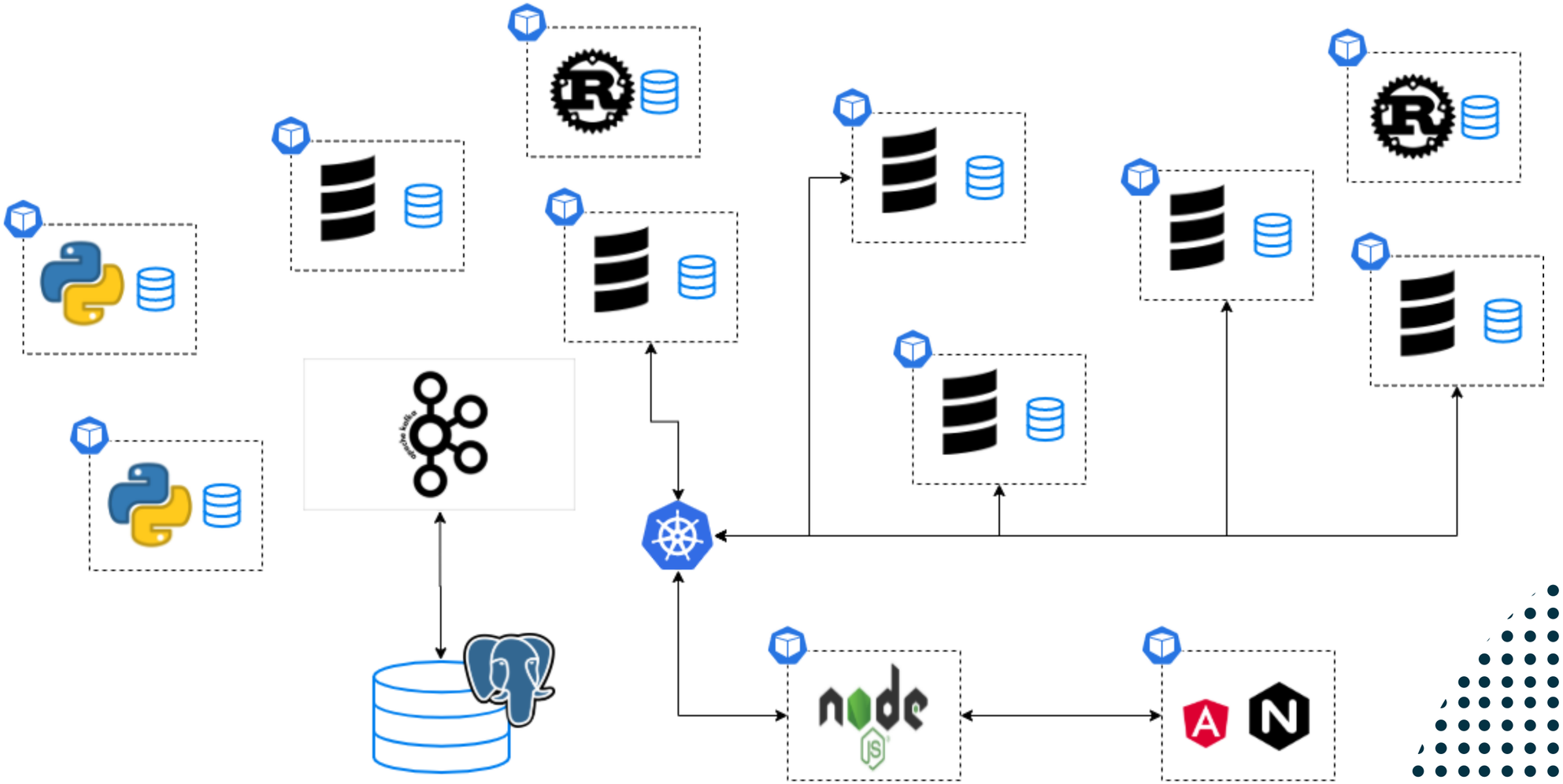
CQRS

Clearly separate concerns while obtaining potentially different read models for each use case

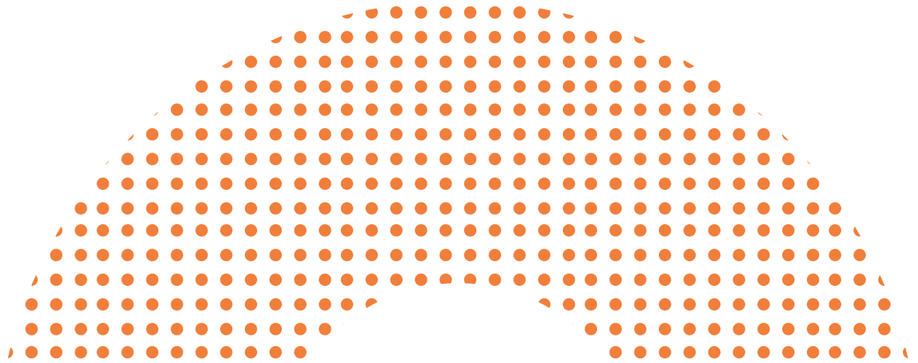
Modernize the stack

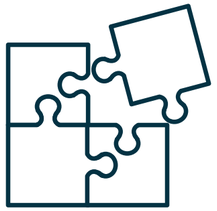
Take advantage of GraphQL to solve our issues with boilerplate, authorization, heavy queries

Credimi 2.1



Exposing the read side



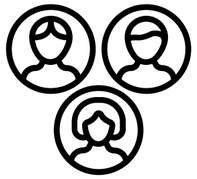


Seamless integration

Dockerized tool easy to set up and to integrate with the existing architecture

Easy to work with

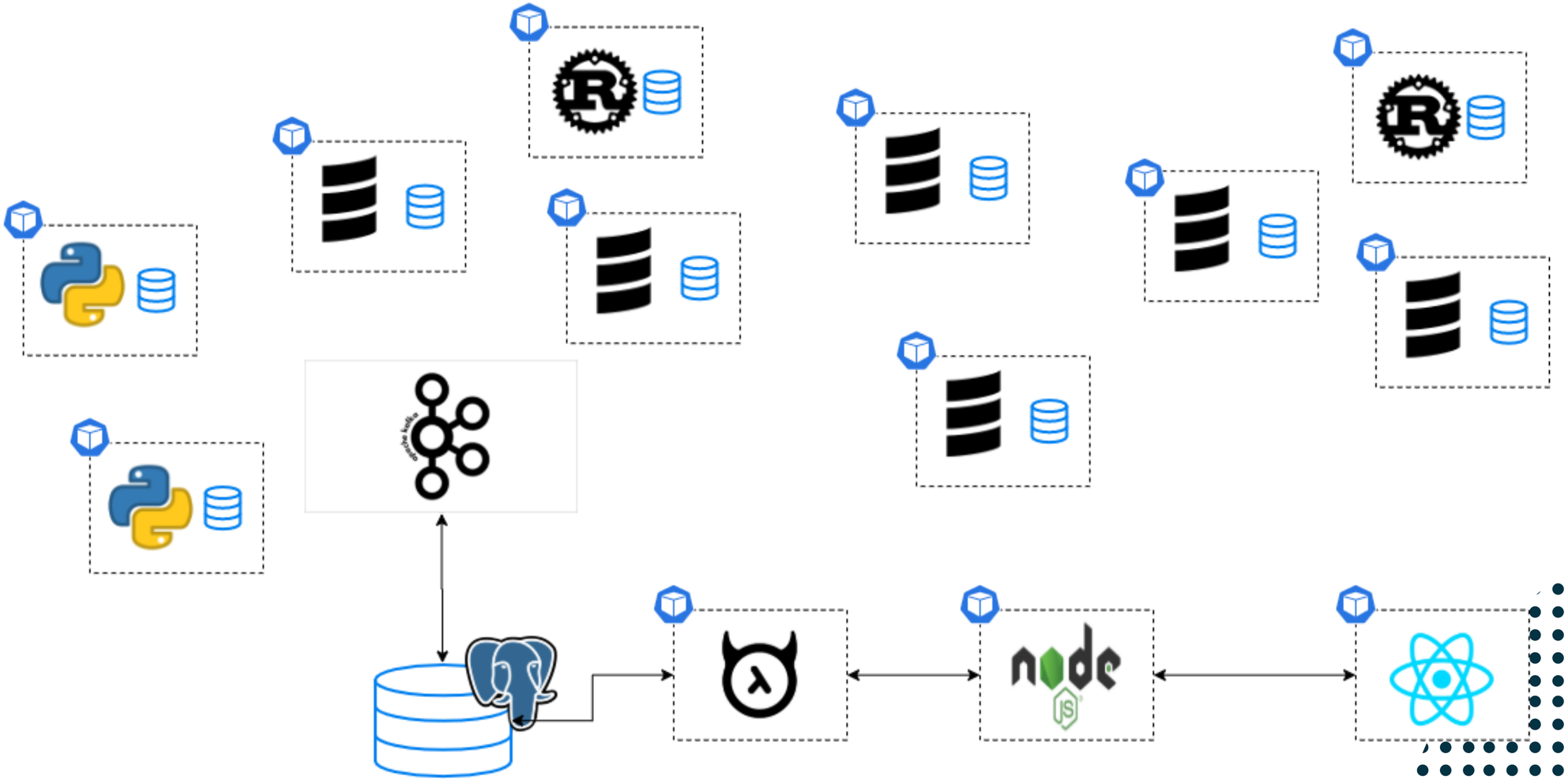
For both backend and frontend



Great community

Transparency and easy to work with

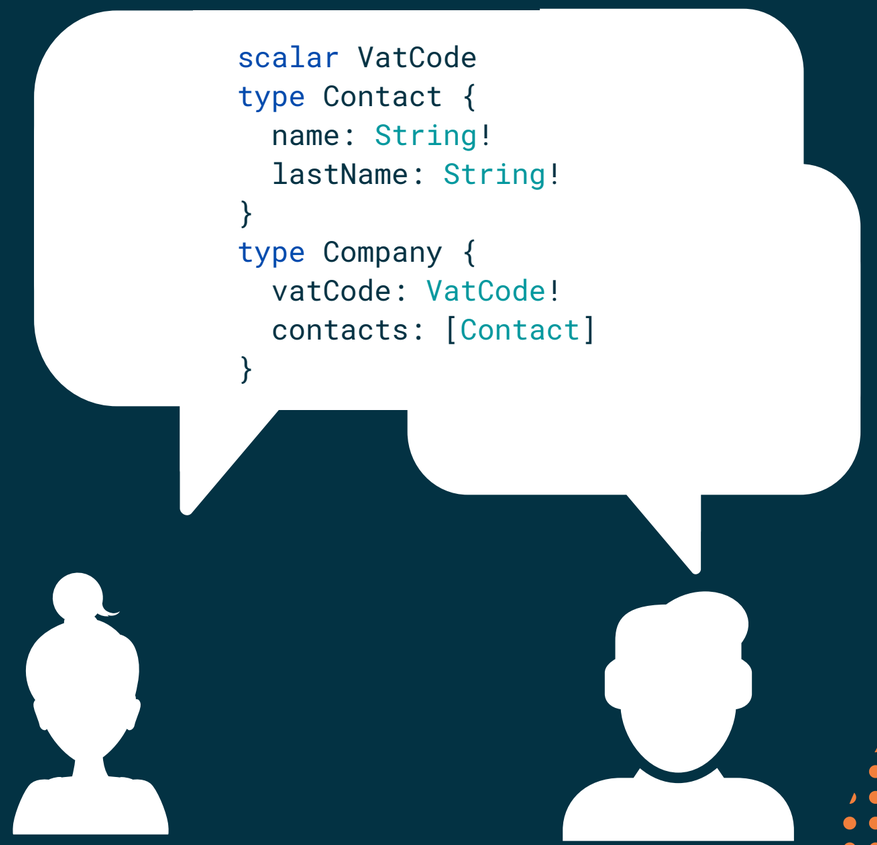
Credimi 3.0



Let's take a look
at the developer
experience now

New feat development

- The developers agree on the resource Graph
  - What are the involved entities
  - Which properties are exposed
  - Who can see those properties



New feat development

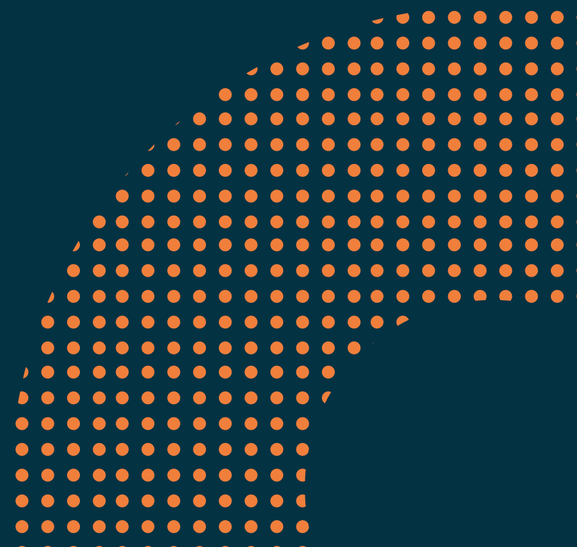
- The developers agree on the resource Graph
- The backend developer brings up a new environment
  - A new namespace on k8s with all the needed microservices
  - A dedicated PostgreSQL database
  - A Hasura instance

New feat development

```
~/read-side » tree
.
├── deployment.yml
├── hasura
│   ├── Dockerfile
│   ├── hasura-migrations
│   │   └── metadata.json
│   └── tests
│       ├── Dockerfile
│       ├── local_build_and_run_env.sh
│       ├── Pipfile
│       ├── Pipfile.lock
│       └── test_authorization.py
└── migrations
    ├── migrations
    │   └── V1__Initial_setup.sql
    └── tests
        ├── Dockerfile
        ├── local_build_and_run_env.sh
        ├── Pipfile
        ├── Pipfile.lock
        └── test_migrations.py
```

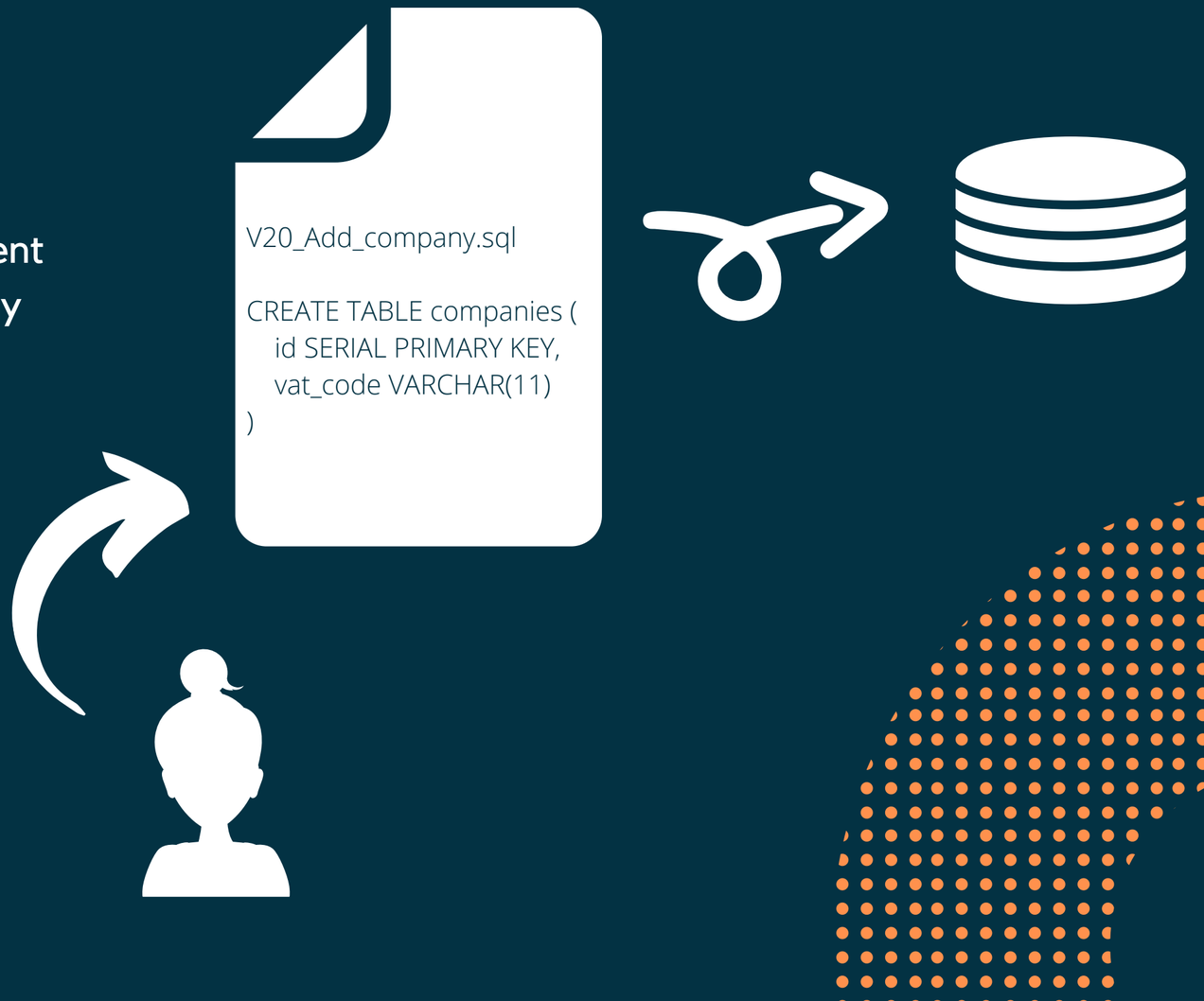
```
~/read-side » cat hasura/Dockerfile
FROM hasura/graphql-engine:v1.2.1.cli-migrations-v2

COPY hasura-migrations /hasura-migrations
```



New feat development

- The developers agree on the resource Graph
- The backend developer brings up a new environment
- The backend developer applies to the database any migration if needed

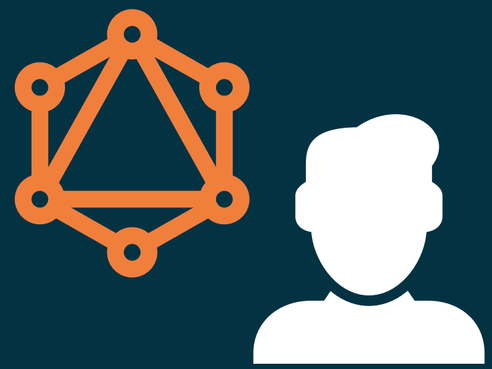


New feat development

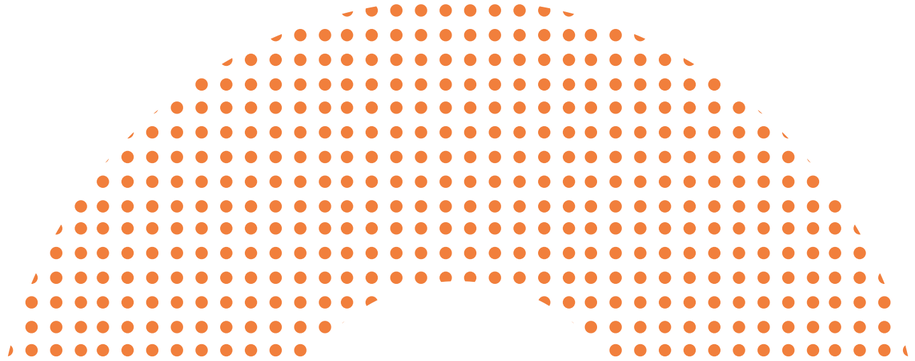
- The developers agree on the resource Graph
- The backend developer brings up a new environment
- The backend developer applies to the database any migration if needed
- The backend developer configure field level authorization on Hasura
  - The metadata are then exported and versioned
  - Every new development can check the validity of both migrations and authorization

New feat development

- The developers agree on the resource Graph
- The backend developer brings up a new environment
- The backend developer applies to the database any migration if needed
- The backend developer configure field level authorization on Hasura
- The frontend developer can now start developing without mocks
  - They can plug Apollo to the running Hasura to fetch the schema
  - If anything needs to change they can act on Hasura UI and then put those changes under versioning
- The backend developer can focus on more fun stuff



Learnings



Better domain comprehension

The exercise of creation of the Graph has improved our big picture vision

More focus

No boilerplate, focus on delivering value

Time to market improved

Hours, not days, to ship features changes

Performance improved

APIs responds in ms, not seconds

Authorization improved

Easy control on who can access what

Stay in touch

@pamela_gotti

pamela.gotti@credimi.com