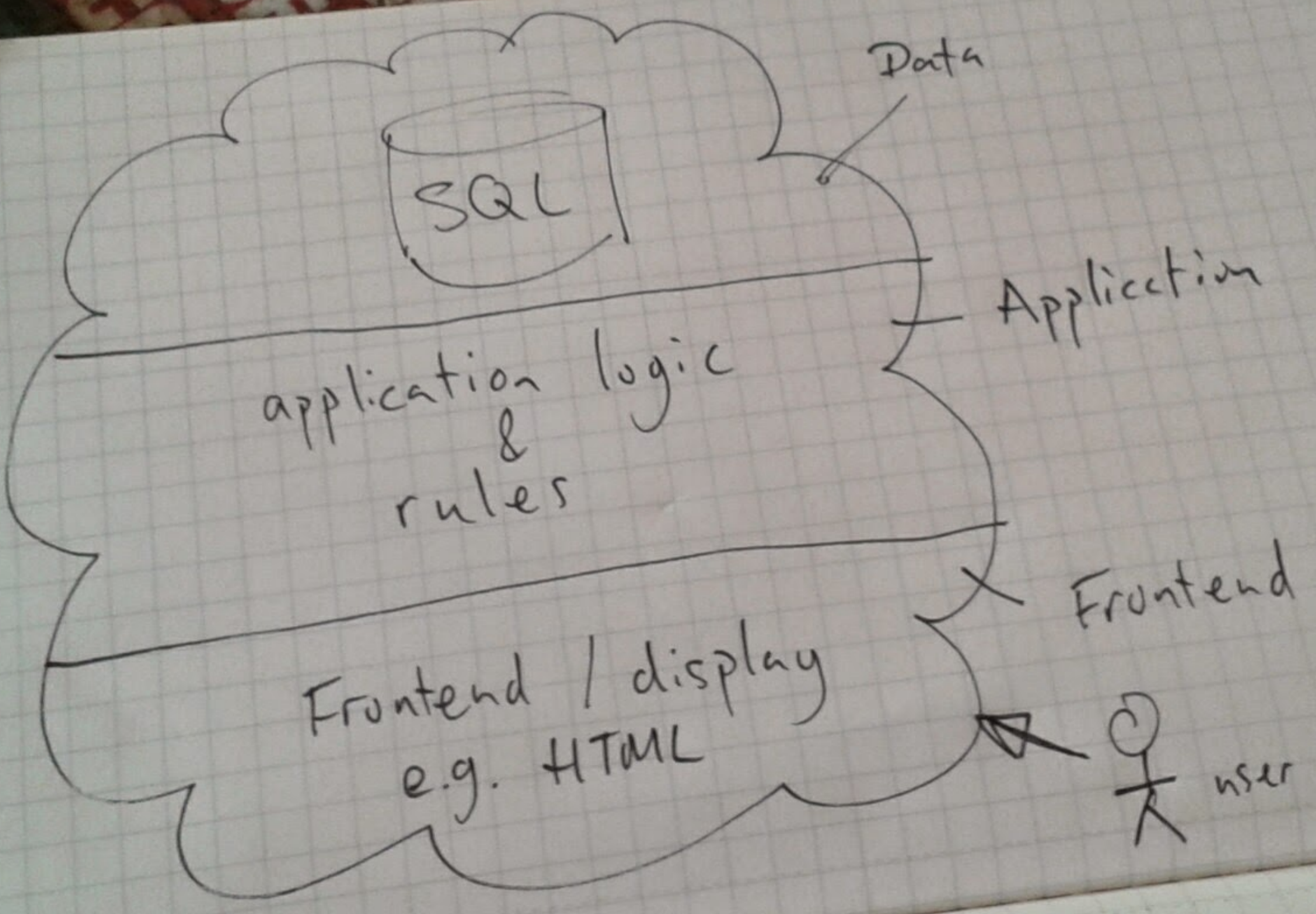
Apps vs. APIs

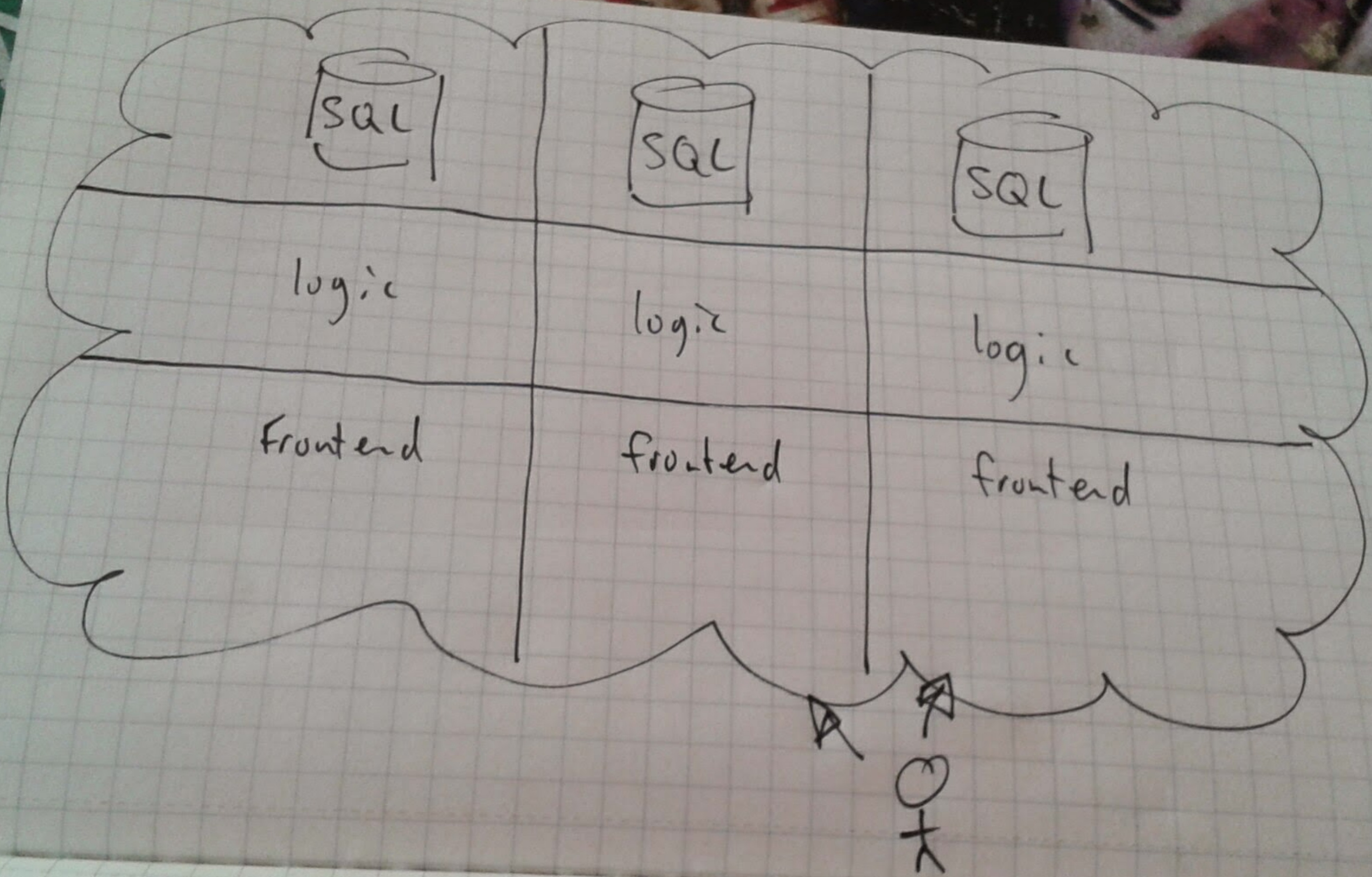
Petrus Janse van Rensburg
PoplusCon, Chile, 2014

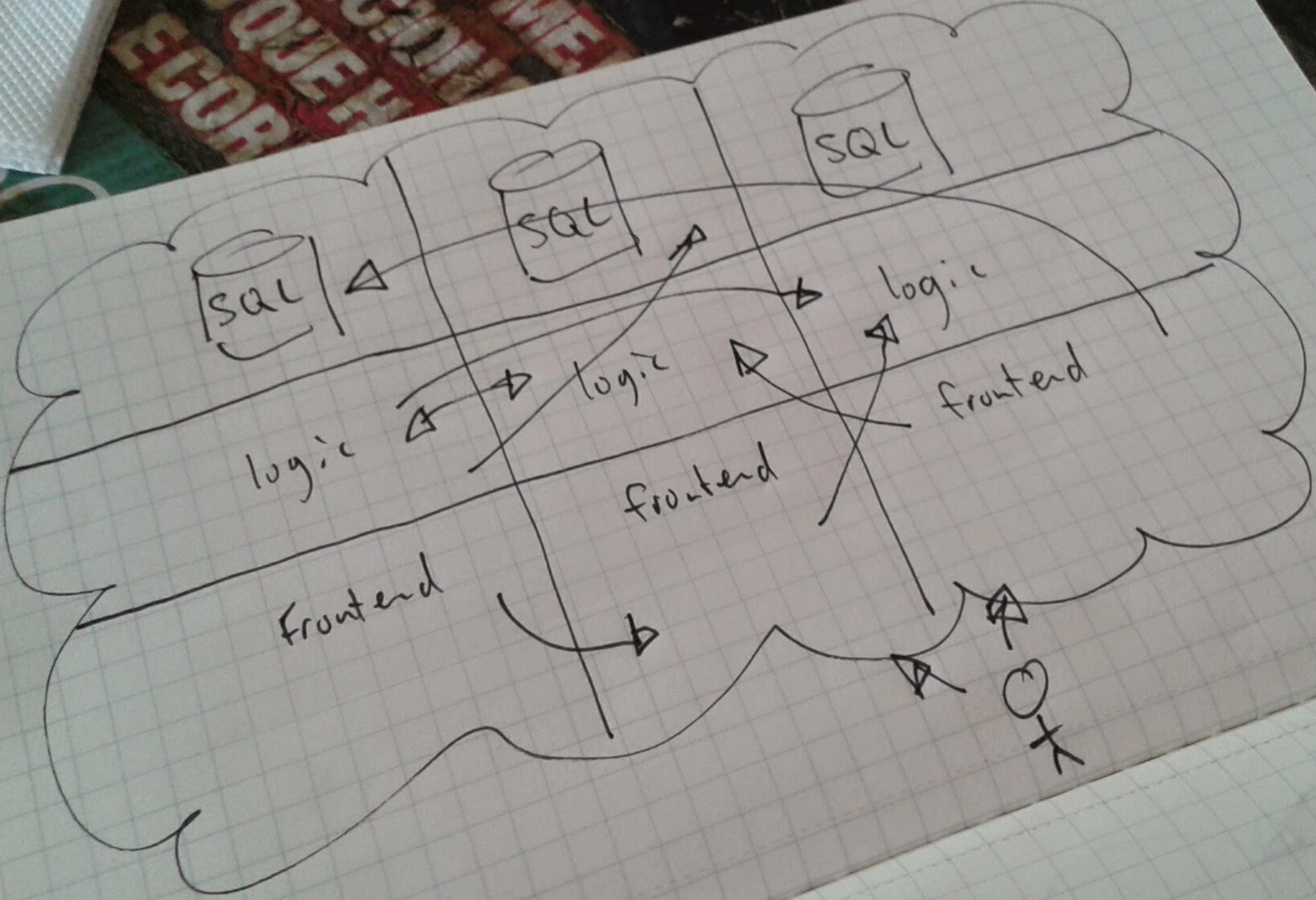
CODE for SOUTH AFRICA

Applications

Typical structure


Data
SQL
Application
application logic & rules
Frontend
Frontend / display e.g. HTML
user


SQL
SQL
SQL
logic
logic
logic
frontend
frontend
frontend


SQL
SQL
SQL
logic
logic
logic
frontend
frontend
frontend

CODE for SOUTH AFRICA

Applications

Pitfalls of the typical structure

Tightly coupled

Large code-bases

Difficult to adapt

Applications

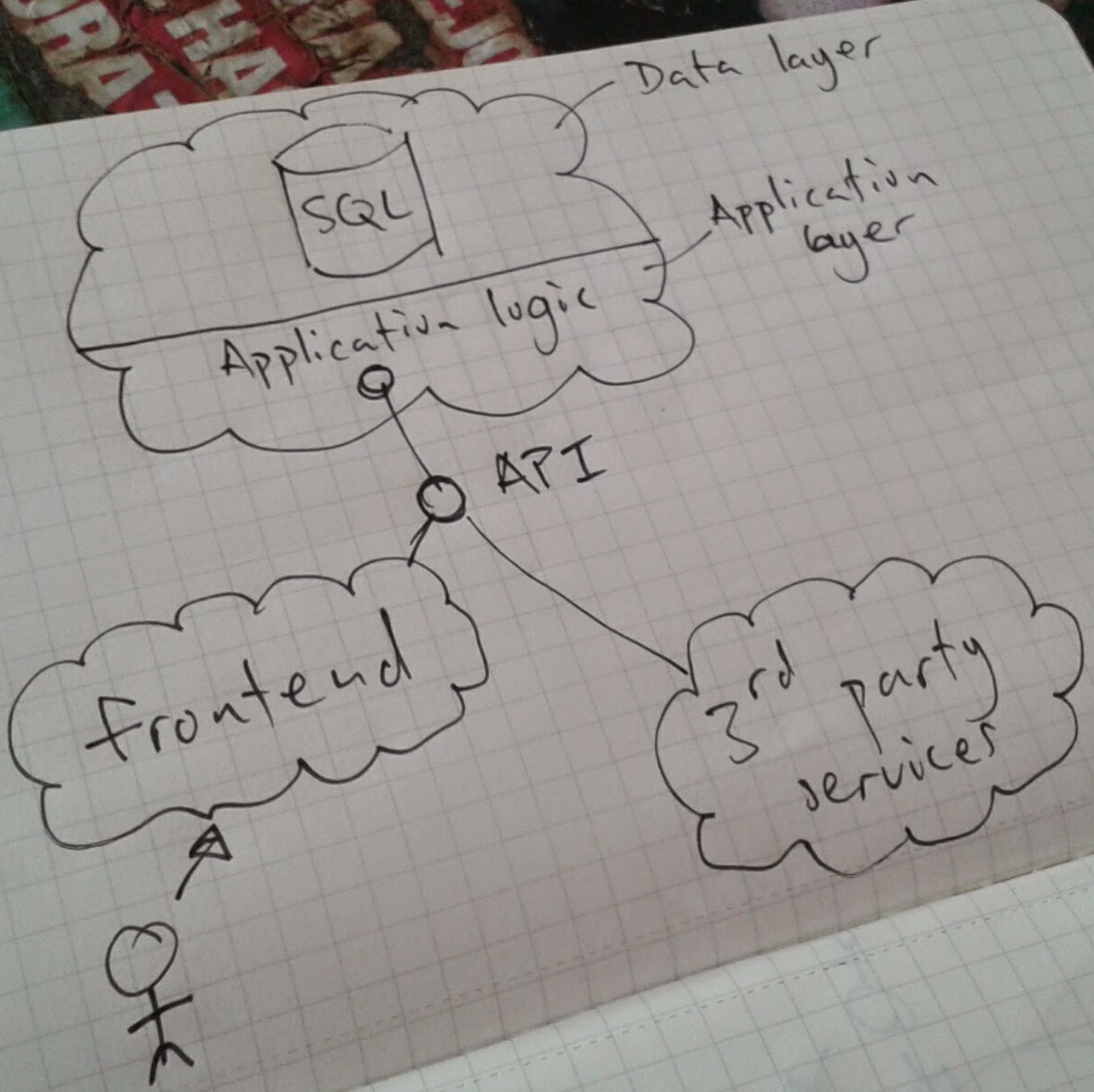
“It’s tough to make predictions, especially about the future”
- Neils Bohr

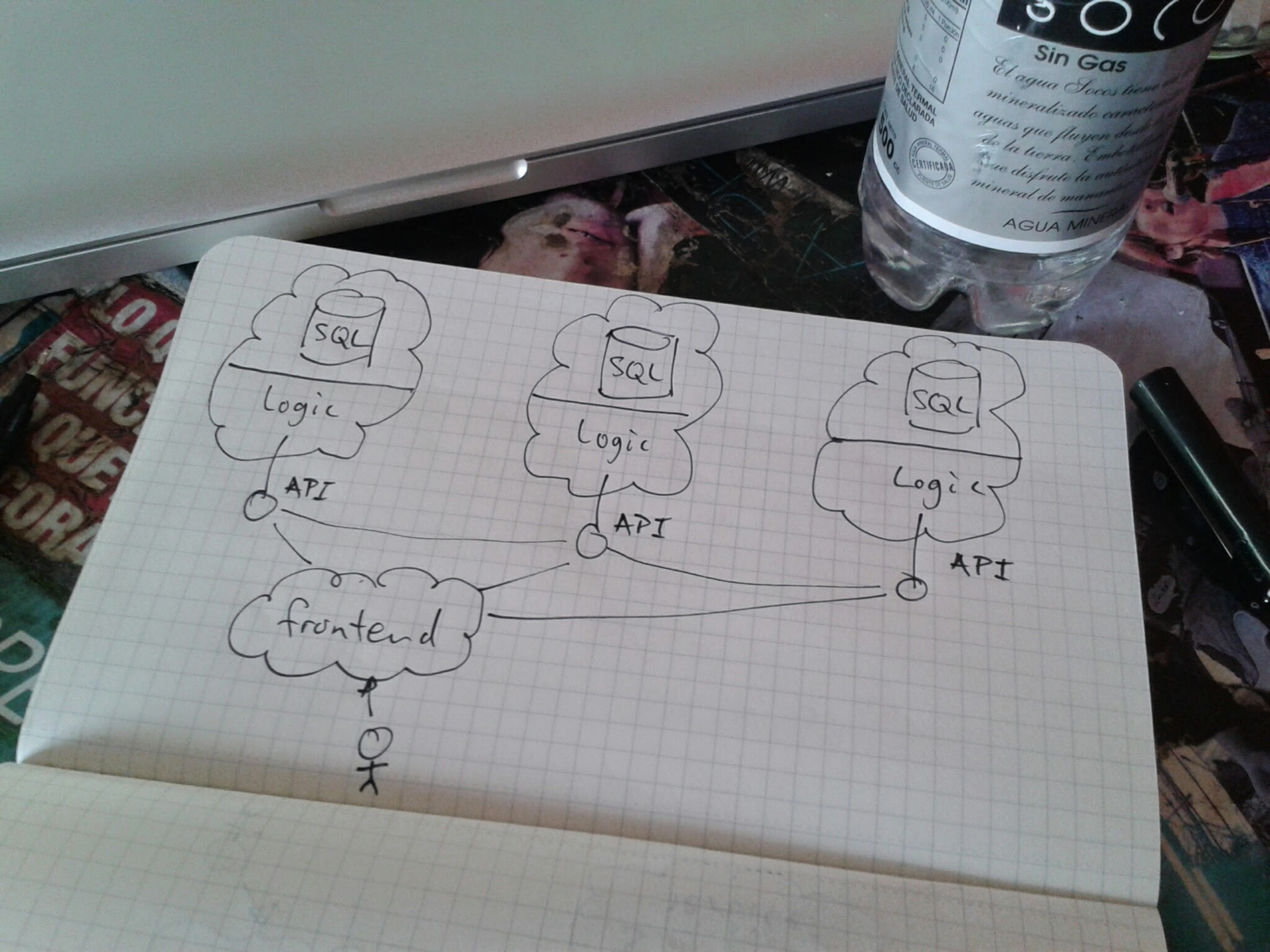
APIs

What is an API?

Application **P**rogramming **I**nterface

Allow software components to interact with each other


Data layer
SQL
Application layer
Application logic
API
frontend
3rd party services


SQL
Logic
API
SQL
Logic
API
SQL
Logic
API
frontend
Sin Gas
AGUA MINERAL

Why APIs?

De-couple components

Easy to re-use code

APIs are language agnostic

Projects are more adaptable

Why APIs?

“If you're not building APIs you're doing it wrong”
- Armin Ronacher

Why APIs?

The Jeff Bezos memo (circa ~ 2002)

1) All teams will henceforth expose their data and functionality through service interfaces.

2) Teams must communicate with each other through these interfaces.

3) There will be no other form of interprocess communication allowed: no direct linking, no direct reads of another team's data store, no shared-memory model, no back-doors whatsoever. The only communication allowed is via service interface calls over the network.

4) It doesn't matter what technology they use. HTTP, Corba, Pubsub, custom protocols -- doesn't matter. Bezos doesn't care.

5) All service interfaces, without exception, must be designed from the ground up to be externalizable. That is to say, the team must plan and design to be able to expose the interface to developers in the outside world. No exceptions.

6) Anyone who doesn't do this will be fired.

CODE for SOUTH AFRICA

What makes a good API?

1. Simplicity

“Perfection is achieved, not when there is nothing more to add,
but when there is nothing left to take away.”
- Antoine de Saint-Exupery

CODE for SOUTH AFRICA

What makes a good API?

2. Easy to understand

"Think like a wise man but communicate in the language of the people."
- William Butler Yeats

What makes a good API?

3. Designed for humans

“Any fool can write code that a computer can understand. Good programmers write code that humans can understand.”
- Martin Fowler

CODE for SOUTH AFRICA

What makes a good API?

For developers:

1. Have a clear starting point
2. Write the README first + allow for auto-discovery
3. Compelling examples
    1. Copy & Paste = Good
    2. embeddable widgets
4. Provide sensible defaults
5. Make it hard to make mistakes

Which API's would be useful?

Popit

Billit

Mapit

Hosted and maintained centrally

Which API's would be useful?

Election data

Census data

Mapping tools

Messaging services (e.g. SMS, USSD)

CODE for SOUTH AFRICA

References

- Making Software: What Really Works, and Why We Believe It
  - Andy Oram & Greg Wilson

- Designing Poetic APIs - Erik Rose