Design API for multi-tier software

Problems, solutions and design patterns.

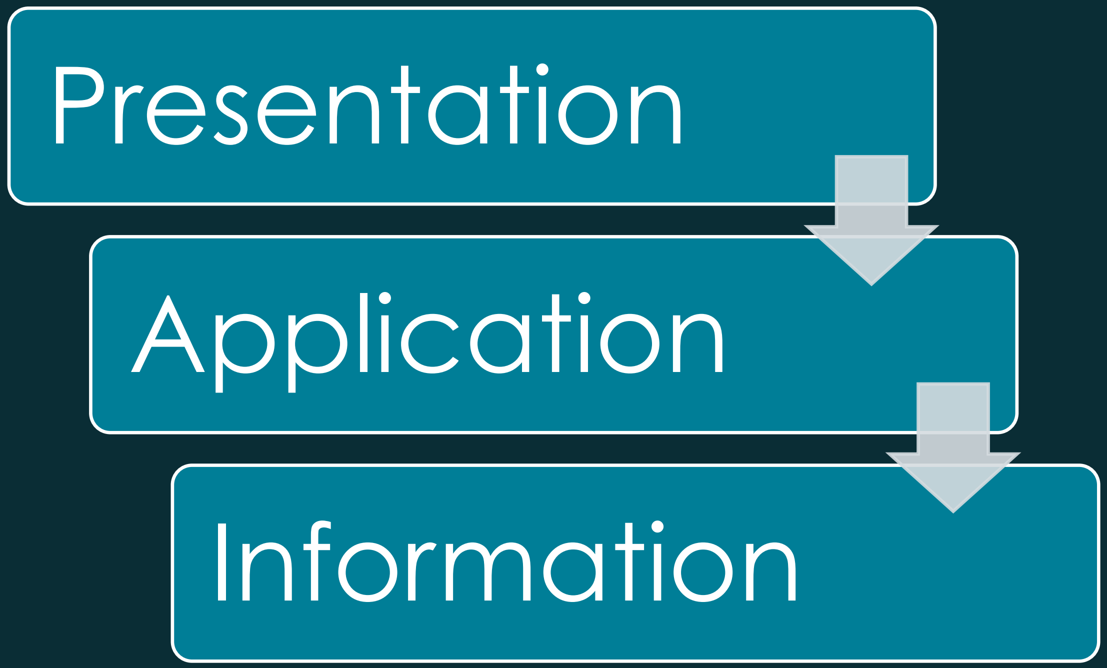
kivio

Marcin (kivio) Karkocha

- Programmer
  - **Python**
  - C#
  - JavaScript
  - From time to time others …
- Speaker
  - PyWAW, PlPycon
  - Targi Kariera IT
- Mentor on Django Carrots
- Blogger in kivio.pl
- Actualy make software for Cyfrowy Polsat S.A.

Multi-tier Software

3-tier software

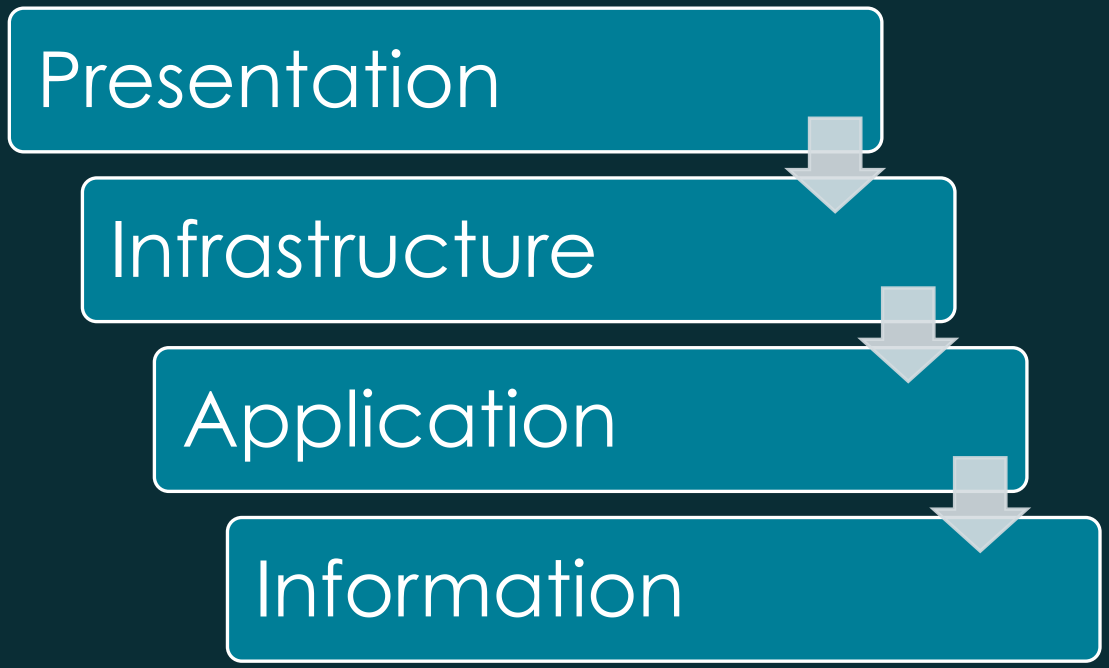


Presentation

Application

Information

4-tier software



Presentation

Infrastructure

Application

Information

Tiers definition

1. **Presentation**
   1. **Clients**
   2. **Web**
   3. **Other application uses our data**
2. Infrastructure
3. **Application**
   1. **Business Logic**
   2. **Data getters**
   3. **Applications**
4. **Data**
   1. **Files**
   2. **Caches**
   3. **DataBases**
   4. **Outsides APIs**

Application

Which structure of application is good for API?

- Model2
- Model View Controller
- Model Template Controller
- …?

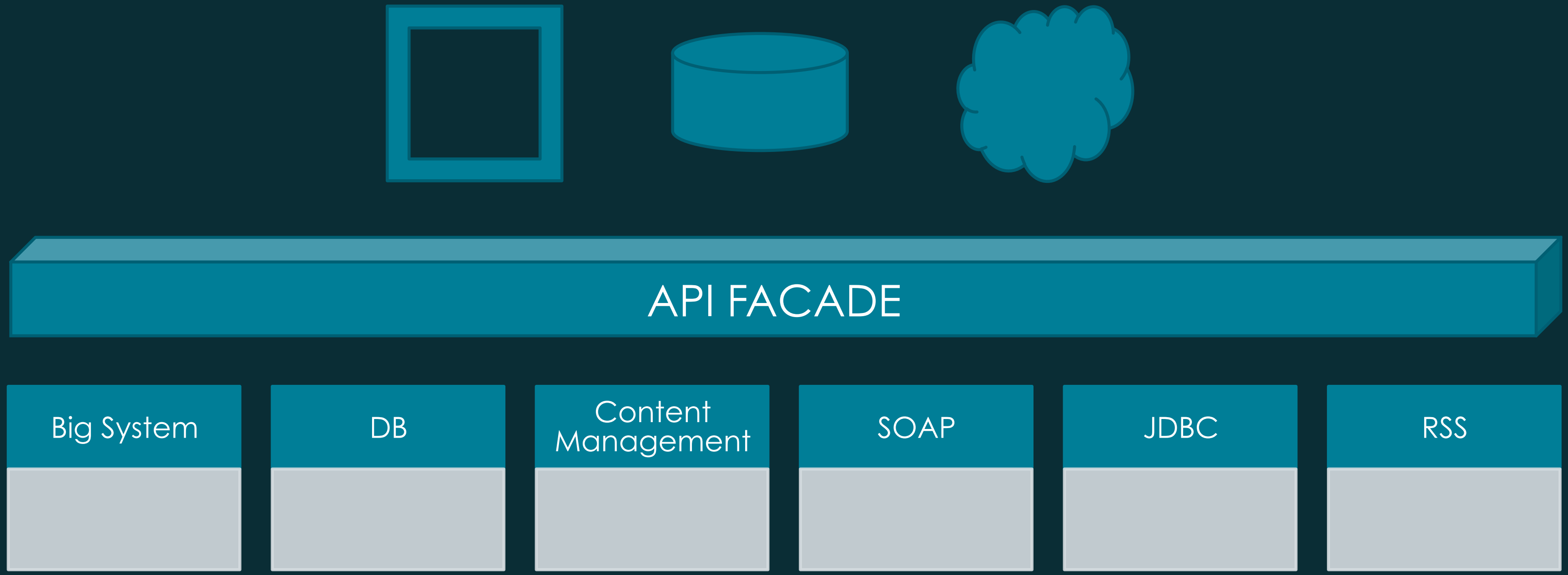
Design Patterns

We don’t need templates (view layer)

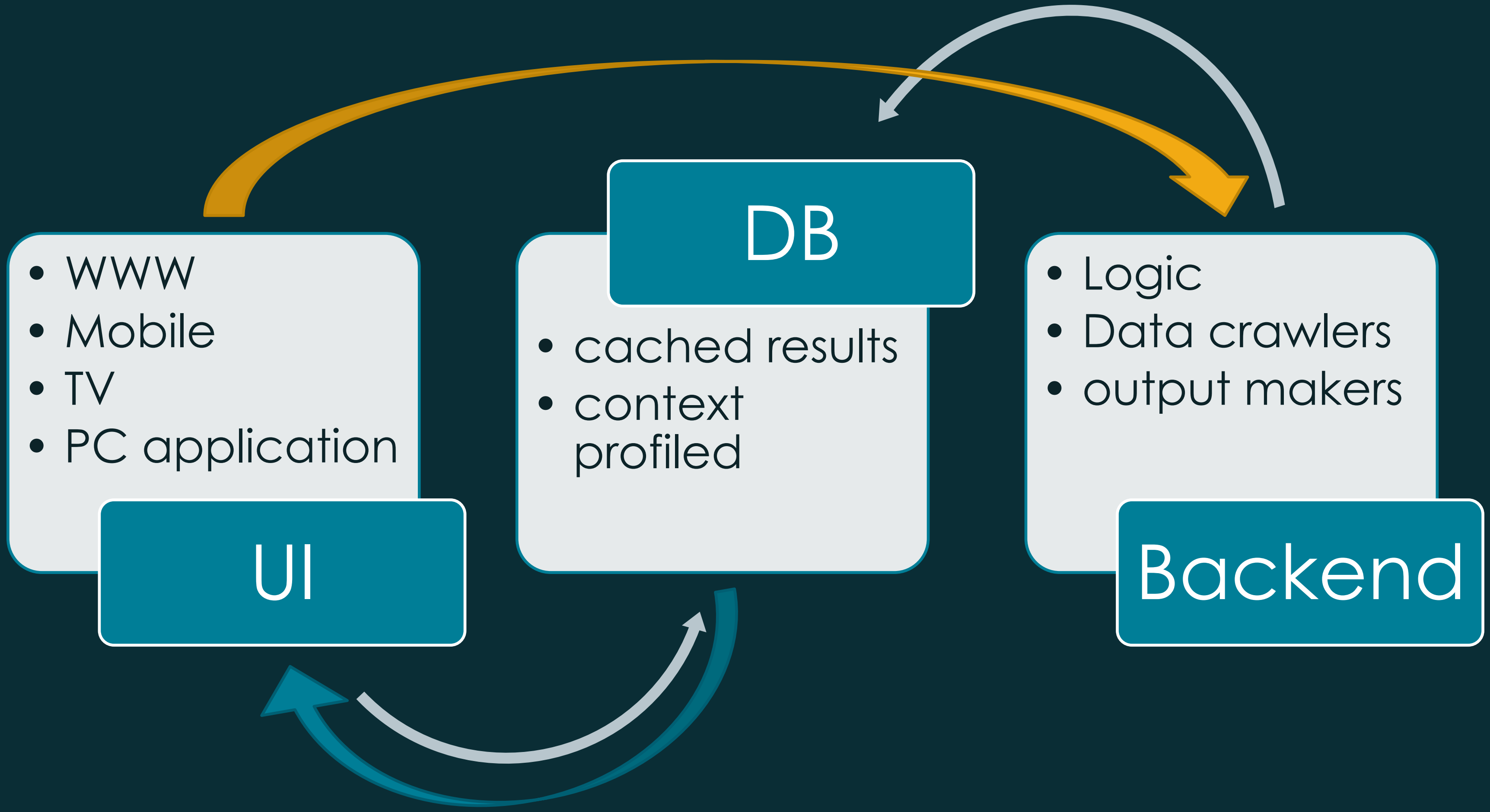
- CQRS *(Command Query Responsibility Segregation)*
- API Fasade
- Monads (Qubes)
- Queue based

Model and Controller

Model = data Tier drivers
Controller -> Design patterns hell



CQRS



- WWW
- Mobile
- TV
- PC application

UI

DB

- cached results
- context profiled

- Logic
- Data crawlers
- output makers

Backend

Model

ORM objects vs. DTO (Data Transfer Object)

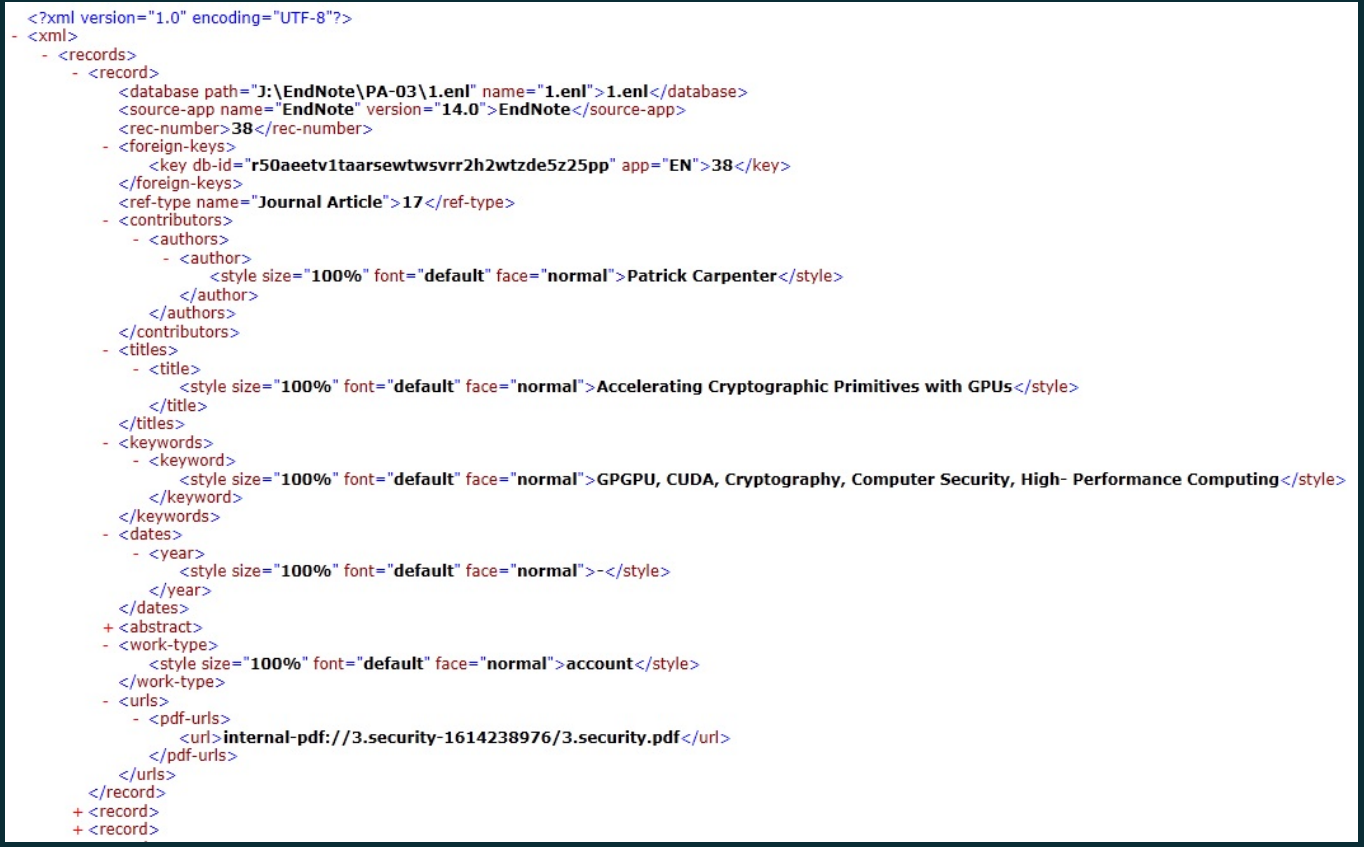
• Easy to make operations • Better understood for Developer • Don’t need additional work • Too close to real data • Not related to BO	• Need more work on start • Need learning on start • One level of astraction higher • Related to Bussines needs

Data serialization

Every API need output format …

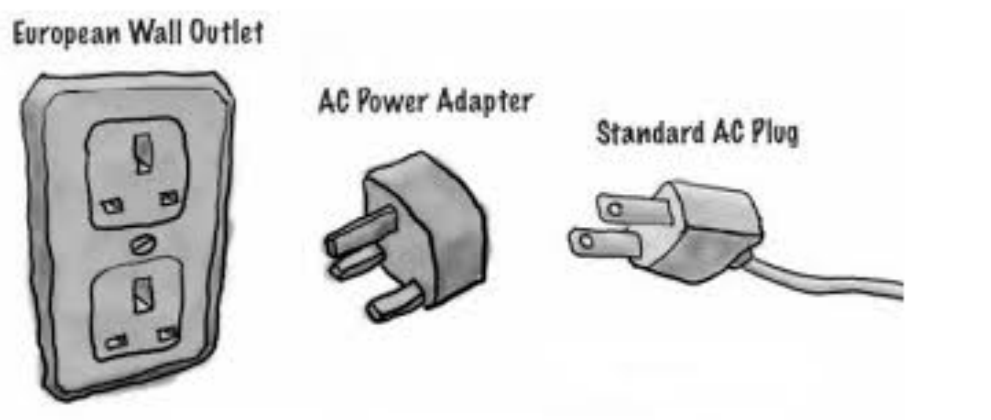
- XML
- JSON
- HTML
- MsgPack
- …

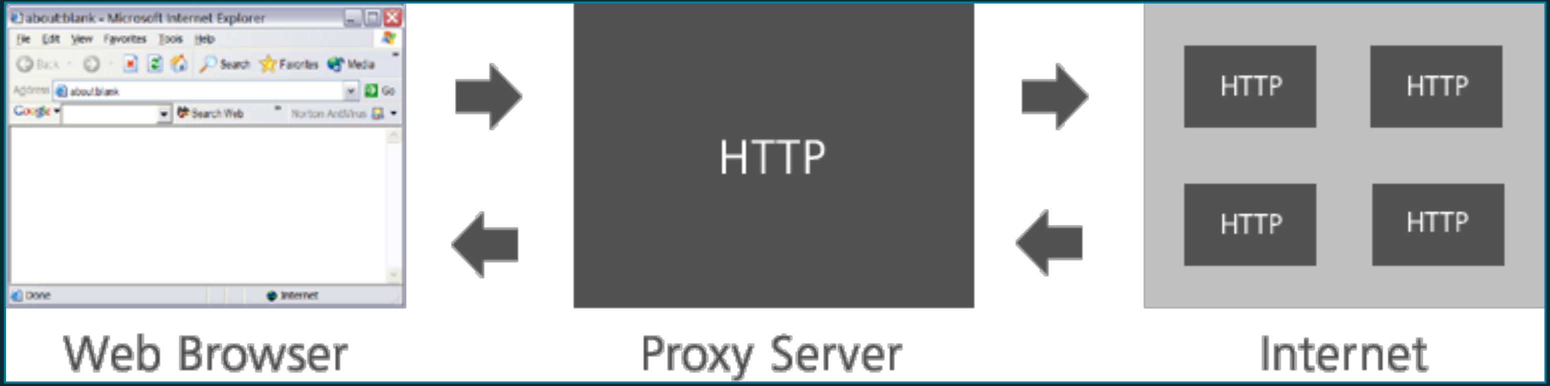
Python object to end of proces …



```
  <?xml version="1.0" encoding="UTF-8"?>
- <xml>
  - <records>
    - <record>
        <database path="J:\EndNote\PA-03\1.enl" name="1.enl">1.enl</database>
        <source-app name="EndNote" version="14.0">EndNote</source-app>
        <rec-number>38</rec-number>
      - <foreign-keys>
          <key db-id="r50aeetv1taarsewtwsvrr2h2wtzde5z25pp" app="EN">38</key>
        </foreign-keys>
        <ref-type name="Journal Article">17</ref-type>
      - <contributors>
        - <authors>
          - <author>
              <style size="100%" font="default" face="normal">Patrick Carpenter</style>
            </author>
          </authors>
        </contributors>
      - <titles>
        - <title>
            <style size="100%" font="default" face="normal">Accelerating Cryptographic Primitives with GPUs</style>
          </title>
        </titles>
      - <keywords>
        - <keyword>
            <style size="100%" font="default" face="normal">GPGPU, CUDA, Cryptography, Computer Security, High- Performance Computing</style>
          </keyword>
        </keywords>
      - <dates>
        - <year>
            <style size="100%" font="default" face="normal">-</style>
          </year>
        </dates>
      + <abstract>
      - <work-type>
          <style size="100%" font="default" face="normal">account</style>
        </work-type>
      - <urls>
        - <pdf-urls>
            <url>internal-pdf://3.security-1614238976/3.security.pdf</url>
          </pdf-urls>
        </urls>
      </record>
    + <record>
    + <record>
```

Proxy vs. Adapter




about:blank - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address about:blank Go
Google Search Web Norton AntiVirus
Done Internet
HTTP
HTTP
HTTP
HTTP
HTTP
Web Browser
Proxy Server
Internet

Cache?

- redis
- memcache
- varnish
- DB
- LiteObjects (ZODB)

Flyweight Design Pattern

Object Player:

- for Rank List
- for Battle
- for Shop
- full object
- …

```
gamer = Player.objects.get(uname=‚cc’)
gamer.fill_for_battle()
gamer.fill_for_shop()
gamer.fill_full()
```

or …

```
gamer.fill_fields([‚name’, ‚rank’, ‚email’, ‚money’])
```

Controller

chain of operations is depends on context

What is context?

- user agent
- user location
- user privilages
- payed packages
- …

Factory of objects

Factory + Strategy

Factory make object for example Video Serializer depends on context (user agent)

Strategy

Factory + Strategy

One scheme of alghoritm with parts implements specific for context

REST vs JSON RPC vs XML RPC … another

Control input data

- wsdl
- json schema
- forms validator
- …

Control output
… same to input

Thanks

I invite you to questions ☺

kivio@kivio.pl
@kivvio on twitter
http://kivio.pl
http://github.com/kivio