
Polyglot Persistence with NoSQL

Advanced software architecture by using multiple
persistence technologies

19. September 2013

Michael Lehmann &
Roman Kuczynski

© Zühlke 2013

```
SELECT      *  
FROM        dbo.Presentation  
WHERE       Title LIKE 'Polyglot pers%'
```

Our holy cow!



One size fits all





Michael Lehmann @lehmamic

Senior Software Engineer @Zühlke since 2012

.Net enterprise and cloud applications

Roman Kuczynski @qtschi

Senior Software Engineer @Zühlke since 2011

Data(base) architectures, BI and Big Data





Borat @BoratNoSQL

Why should I change? It worked for me until now!

Listen to the business





Volume

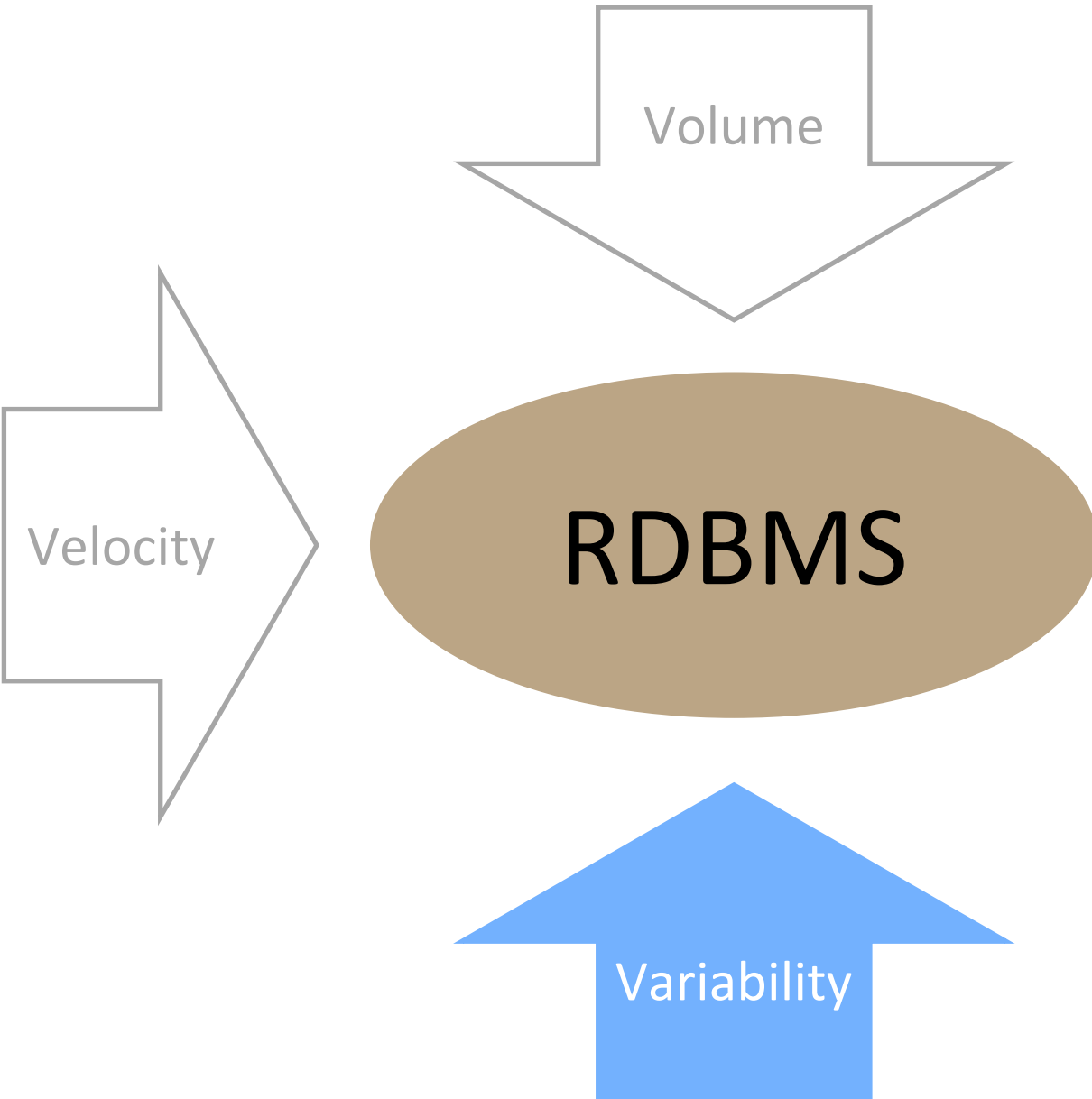


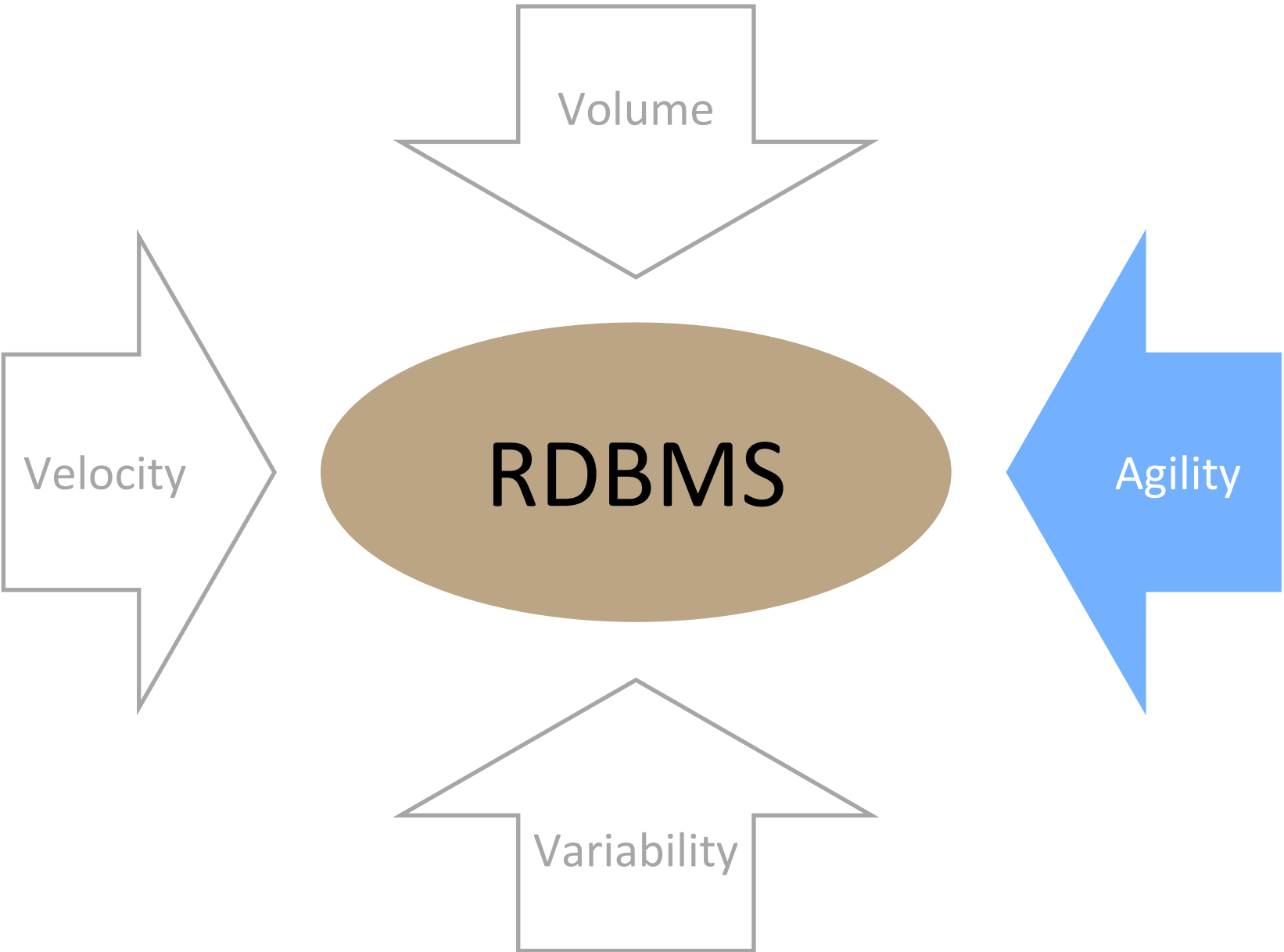
RDBMS

Volume

Velocity

RDBMS





Volume

RDBMS

Agility

Velocity

Variability

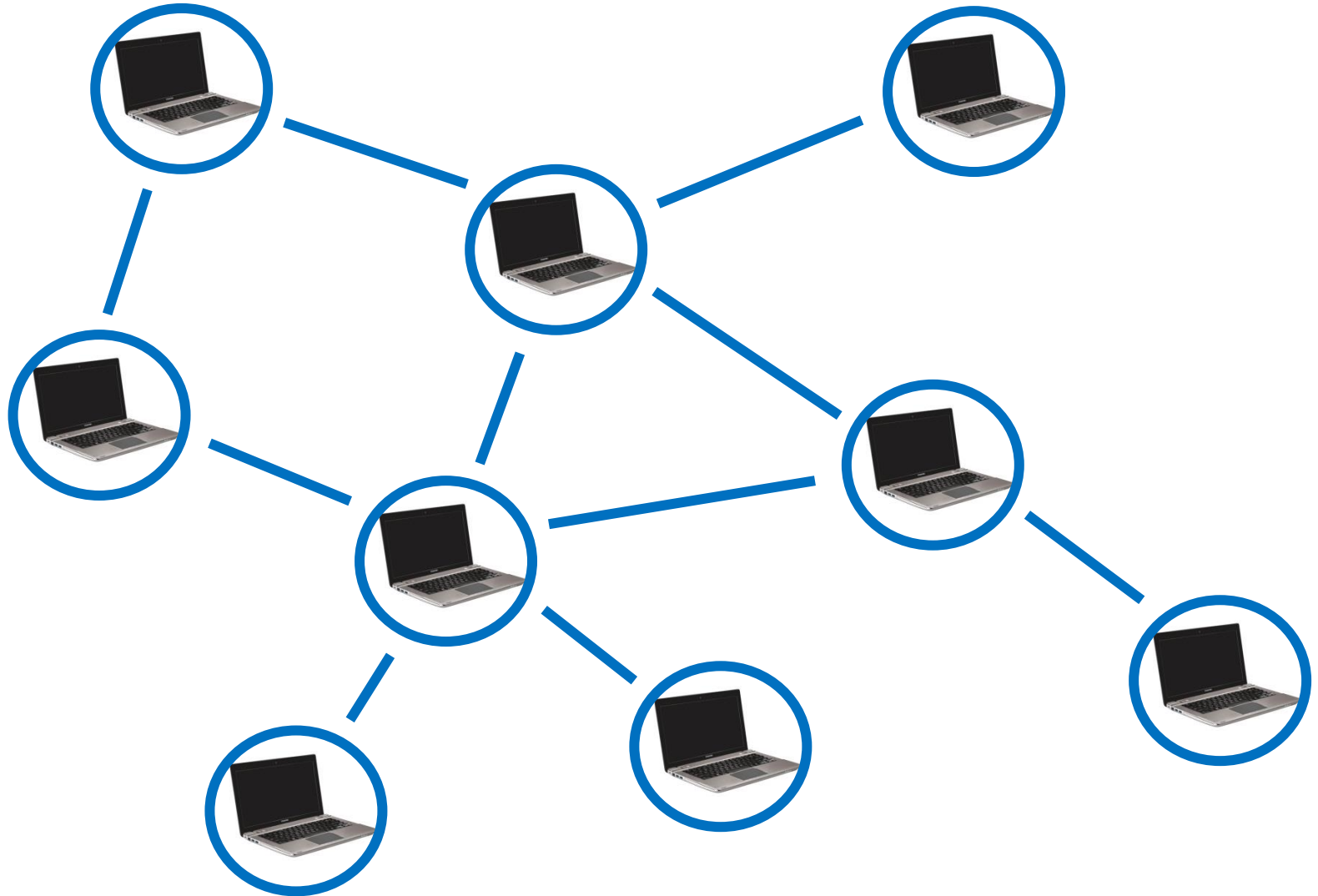


Borat @BoratNoSQL

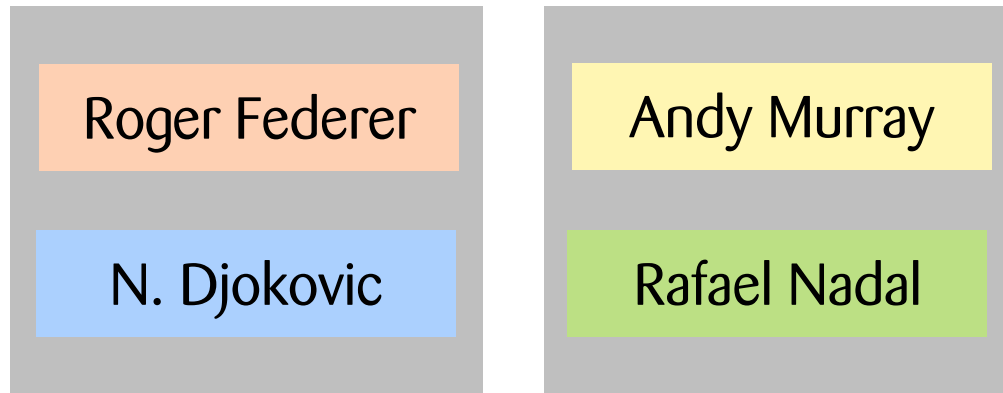
Sounds plausible, but what options do we have?

#NoSQL

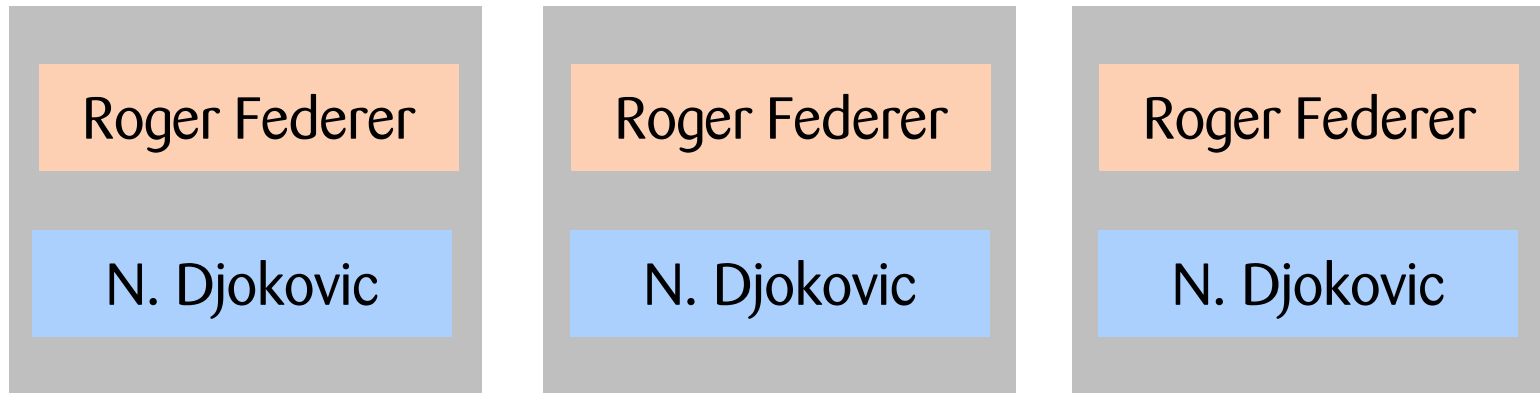
Increasing performance through scale out



Scaling by sharding



Scaling by replication



Impedance mismatch using relational databases

```
public class BlogPost
{
    public int Id { get; set; }
    public string Content { get; set; }
    public List<string> Tags { get; set; }
}
```

BlogPost

- Id (int)
- Content (varchar)

Tag

- Id (int)
- BlogPostId (int)
- Name (varchar)

Design for the relational model

```
public class BlogPost
{
    public int Id { get; set;}
    public List<Tag> Tags { get; set; }
}

public class Tag
{
    public int Id { get; set; }
    public BlogPost BelongsTo { get; set; }
    public string Name { get; set; }
}
```


NoSQL databases increase productivity

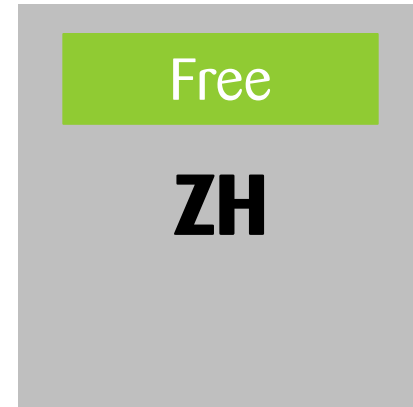
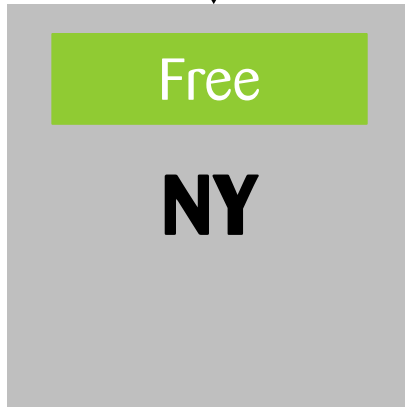
```
var post = new BlogPost
{
    Id = 1,
    Content = "Any text content",
    Tags = new [] { "NoSQL", "Cloud", "PolyglotPersistence" }
};

collection.Insert(post);
```

Data integrity cannot be enforced

NoSQL databases are eventual consistent

We look for a hotel room



We book the room



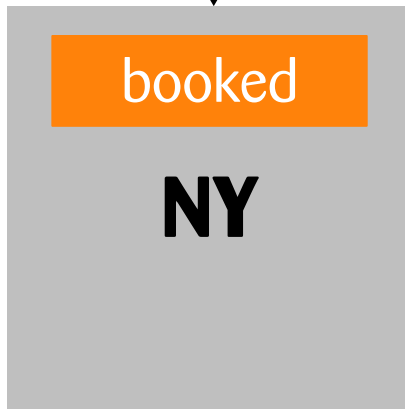
booked

NY

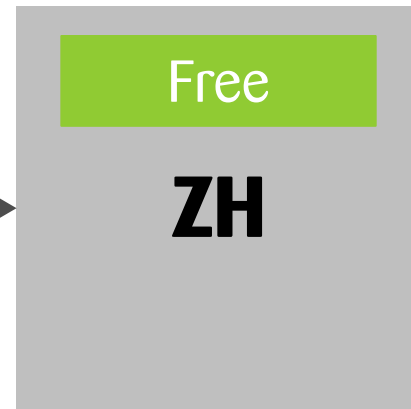
Free

ZH

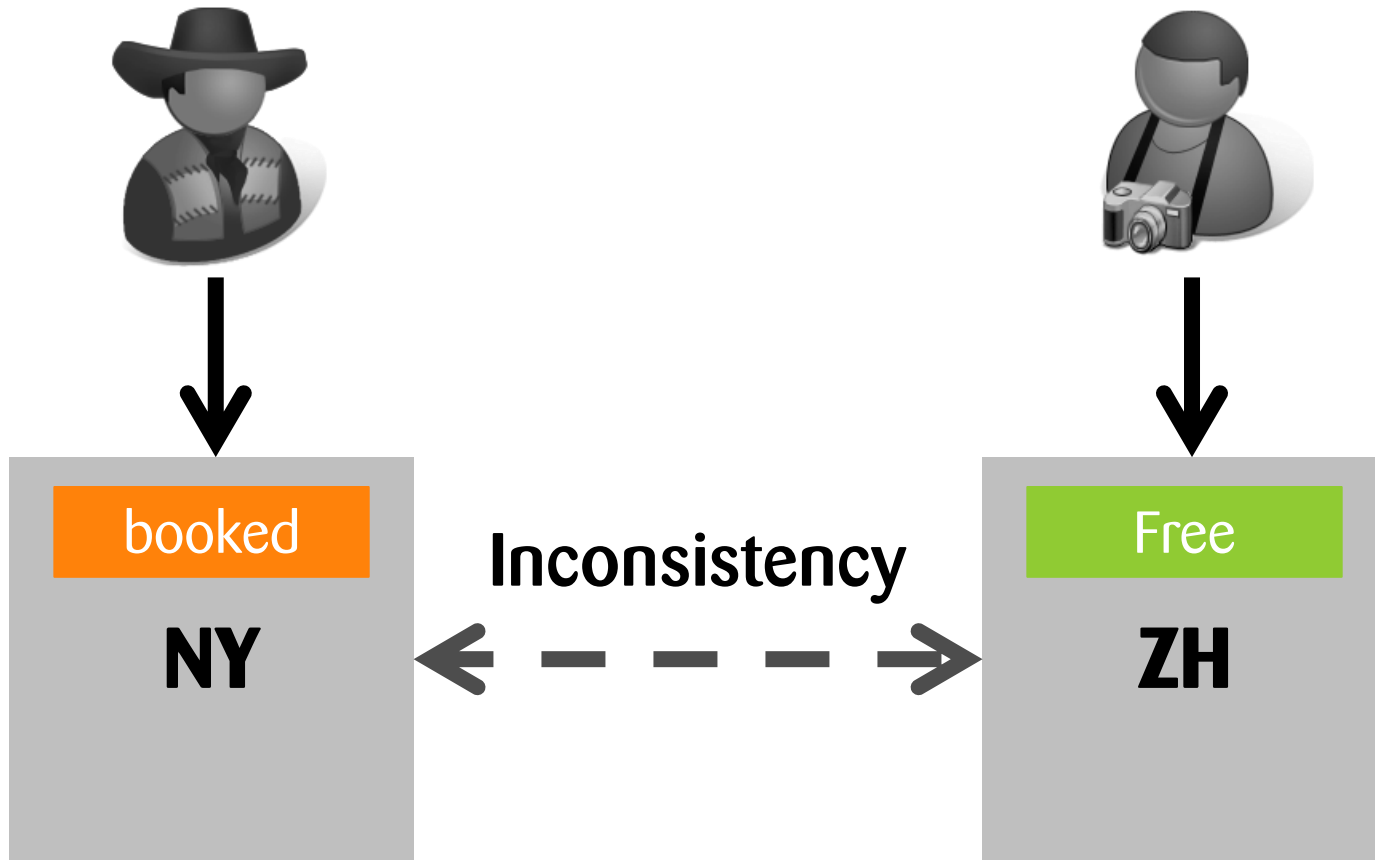
Inconsistency window



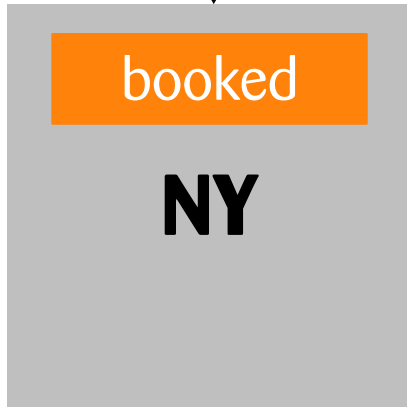
Inconsistency



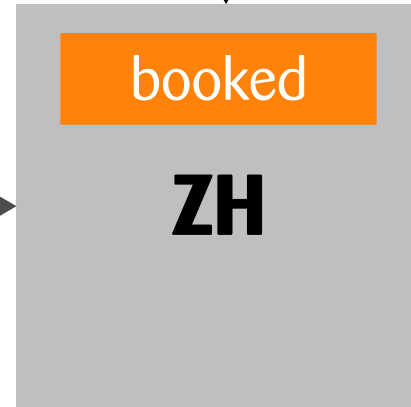
Someone else books the same room



Conflict!



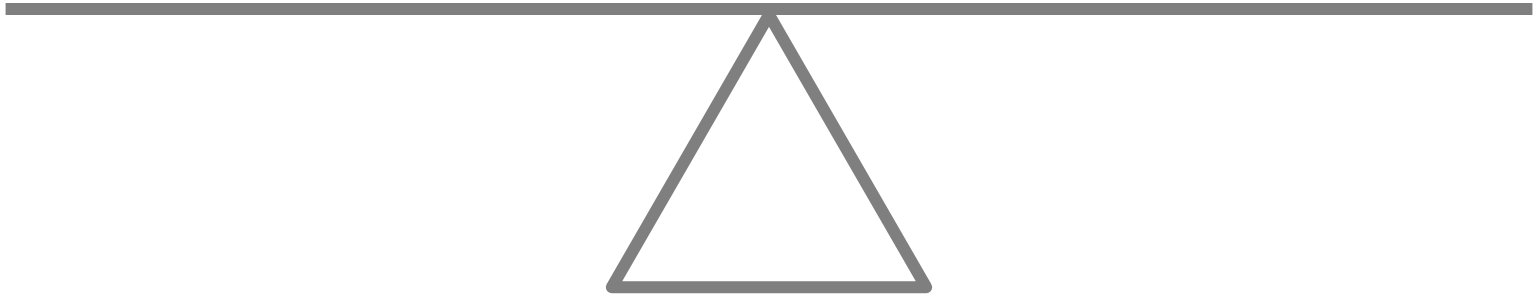
Inconsistency



Why not handle such cases by business?

performance

consistency



What do we have in our toolbox?





A lot of database products



HYPERTABLE INC

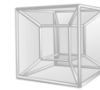


Borat @BoratNoSQL

I feel swamped, how can I differentiate these products?



Key-value stores





Document stores



The document store data model

```
{
  "playerId": 1,
  "firstName": "Roger",
  "lastName": "Federer",
  "ranking": "#1",
  "address": { "city": "Wollerau" }
  "sponsors": [
    {
      "id": 1,
      "name": "Nike"
      "amount": "16'000 SFR"
    },
    {
      "id": 2,
      "name": "Lindt"
      "amount": "5'000 SFR"
    },
    {
      "id": 3,
      "name": "Credit Suisse"
      "amount": "13'000 SFR"
    }
  ]
}
```

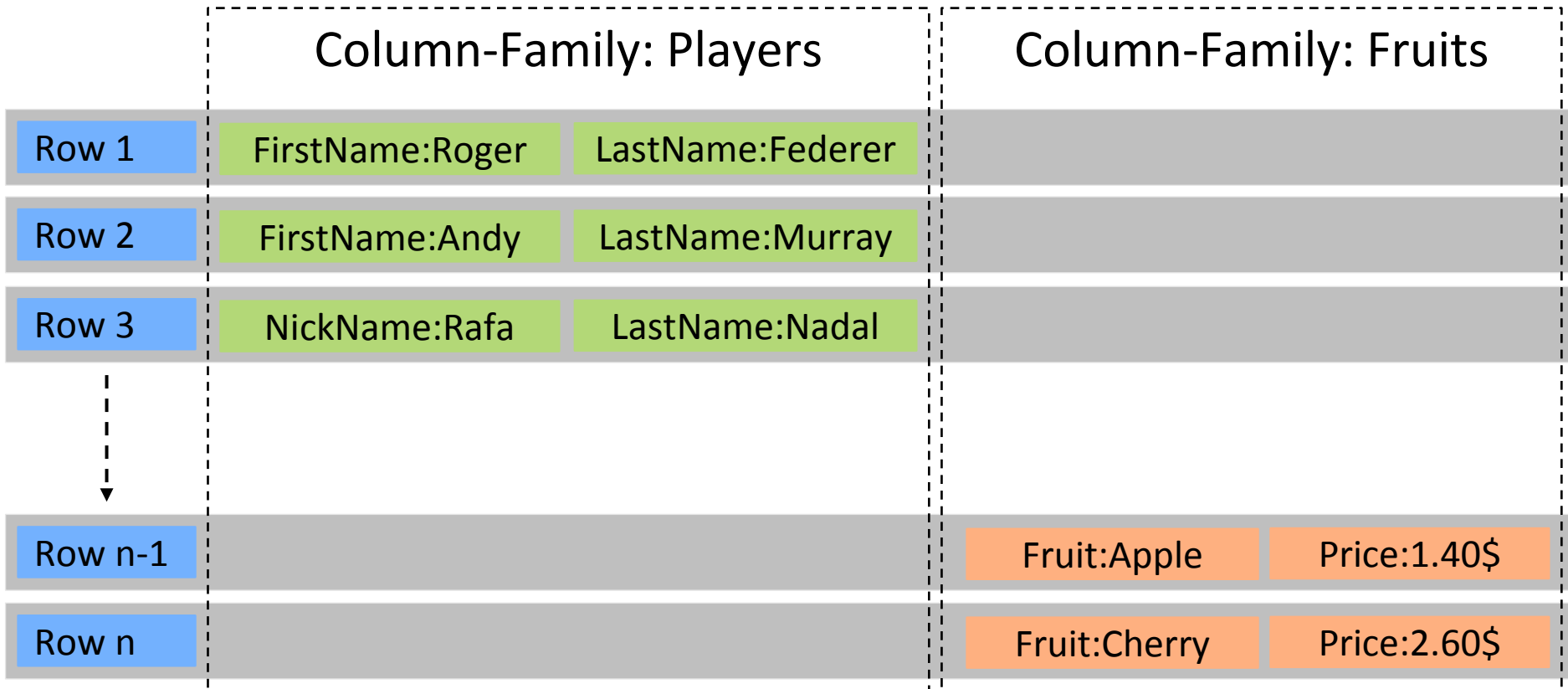


Column-family stores



HYPERTABLE^{INC}

The column-family data model

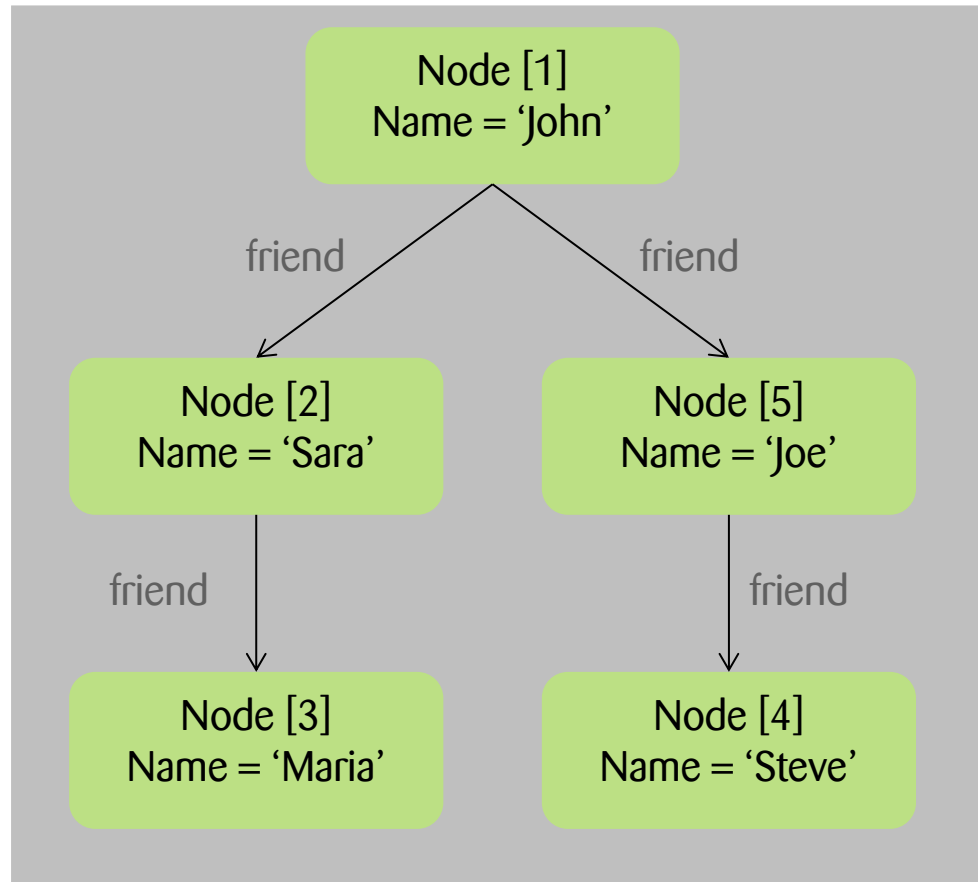




Graph databases

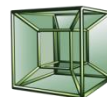


The graph data model





PostgreSQL



HYPERTABLE INC



Decisions, decisions...





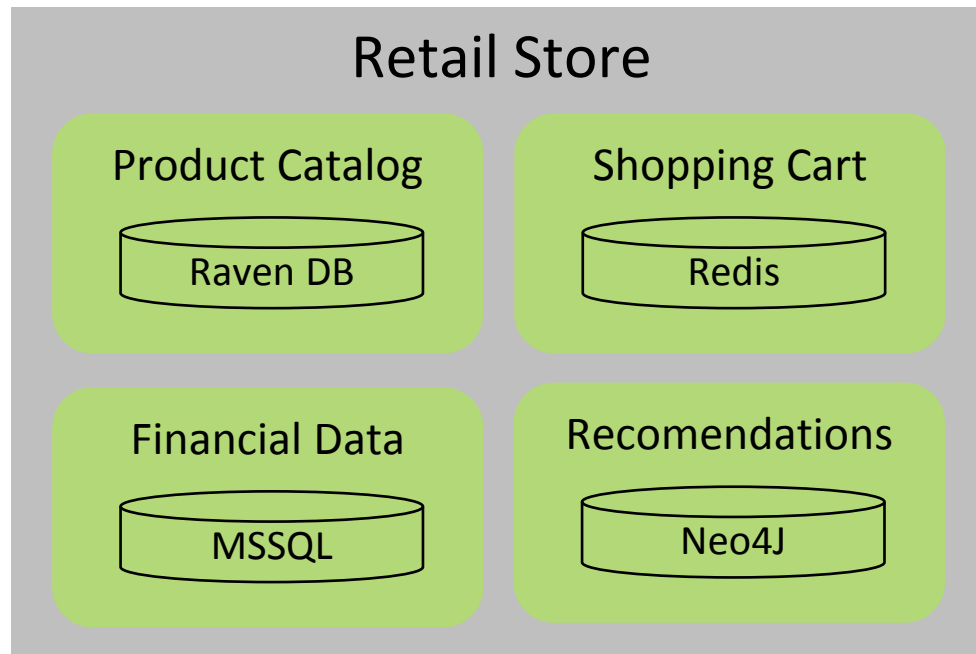
Borat @BoratNoSQL

With every database I have to take tradeoffs into account, I don't want to choose only one!

Pol·y·glot – Adjective
Knowing or using several languages

Per·sist·ence – Noun
The continued or prolonged existence of something

Polyglot persistence illustrated





Borat @BoratNoSQL

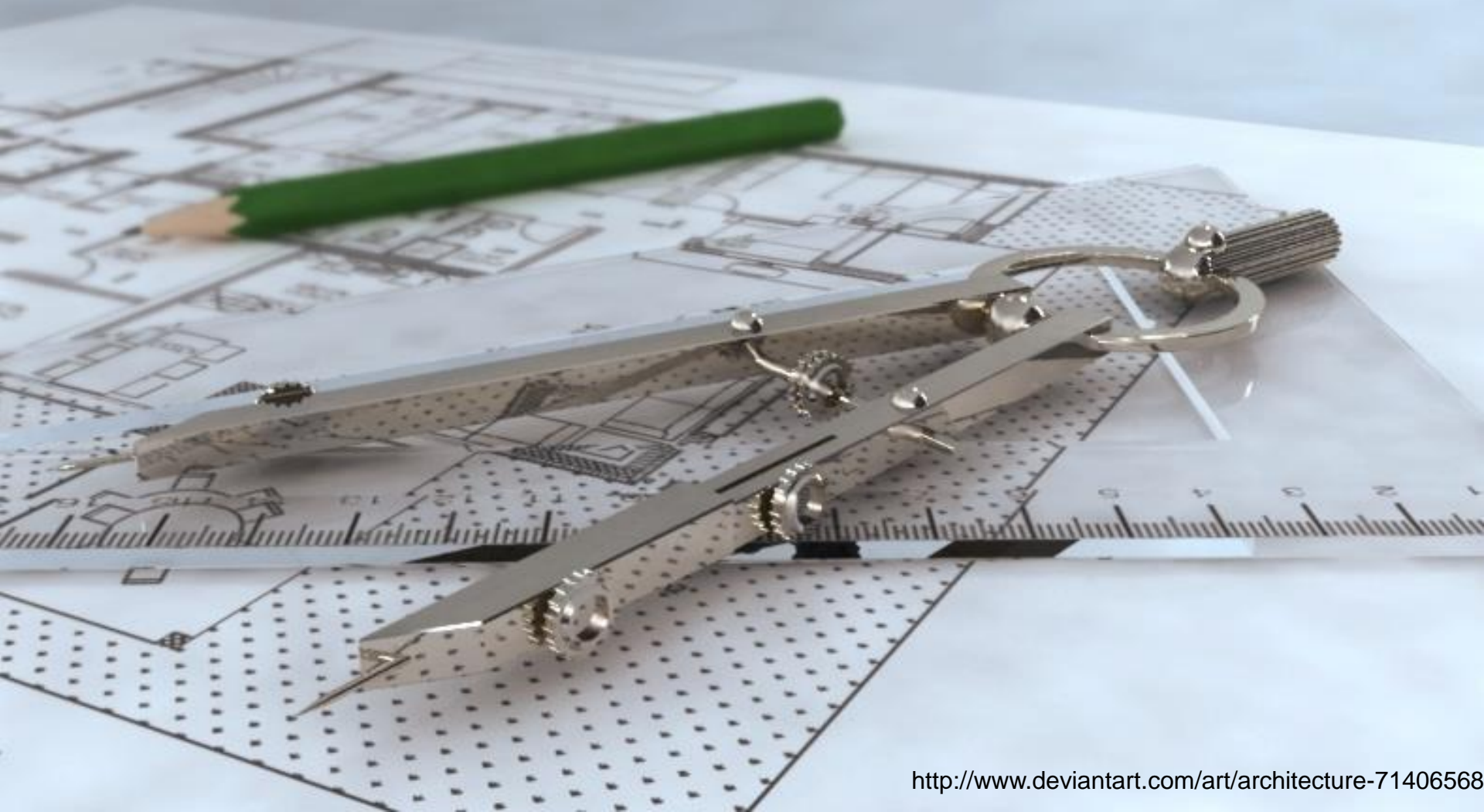
Sounds great! But where is the catch?

We need appropriate skills

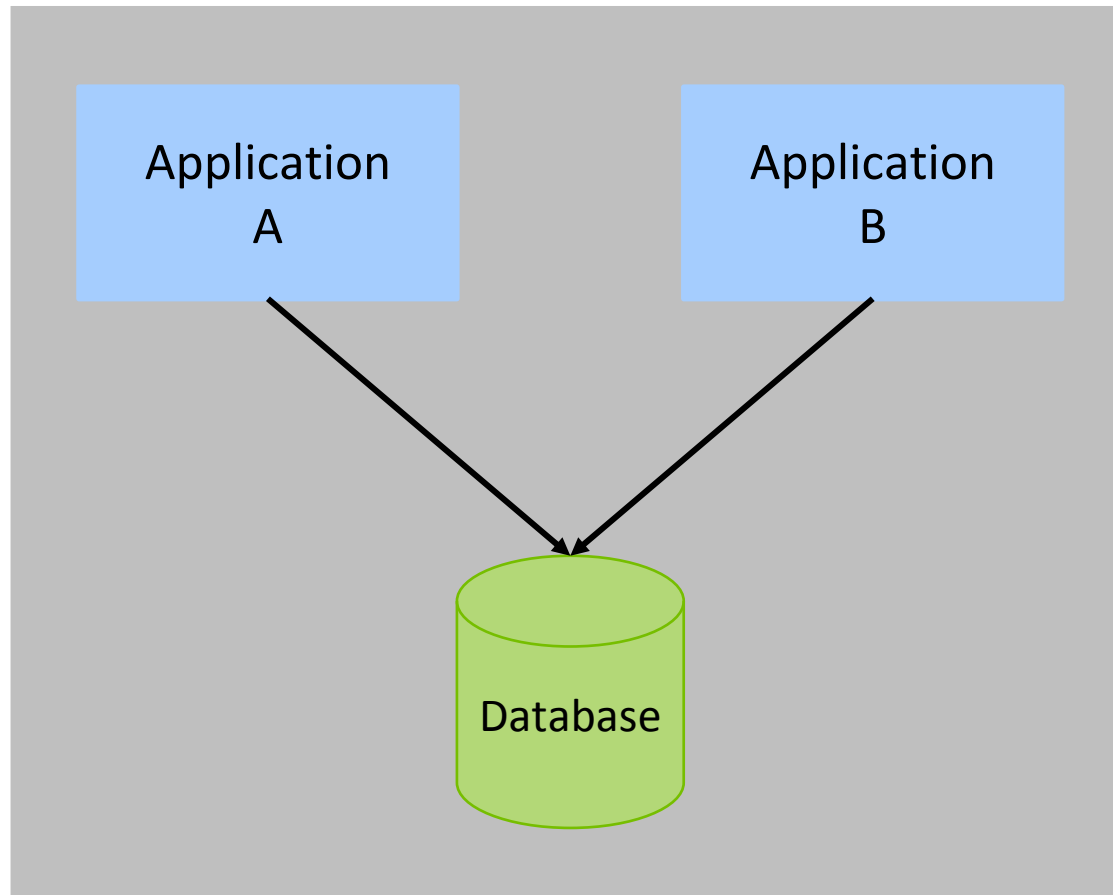
$$E=mc^2$$



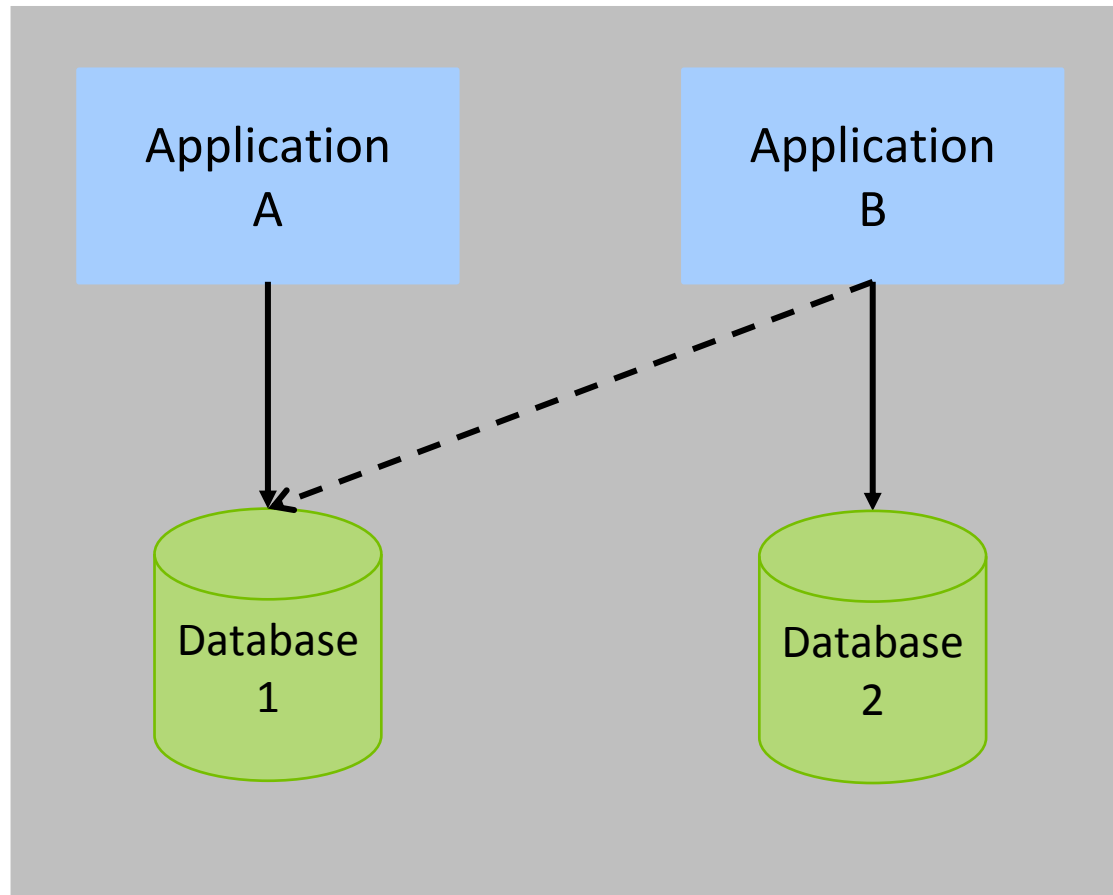
Invest in software architecture



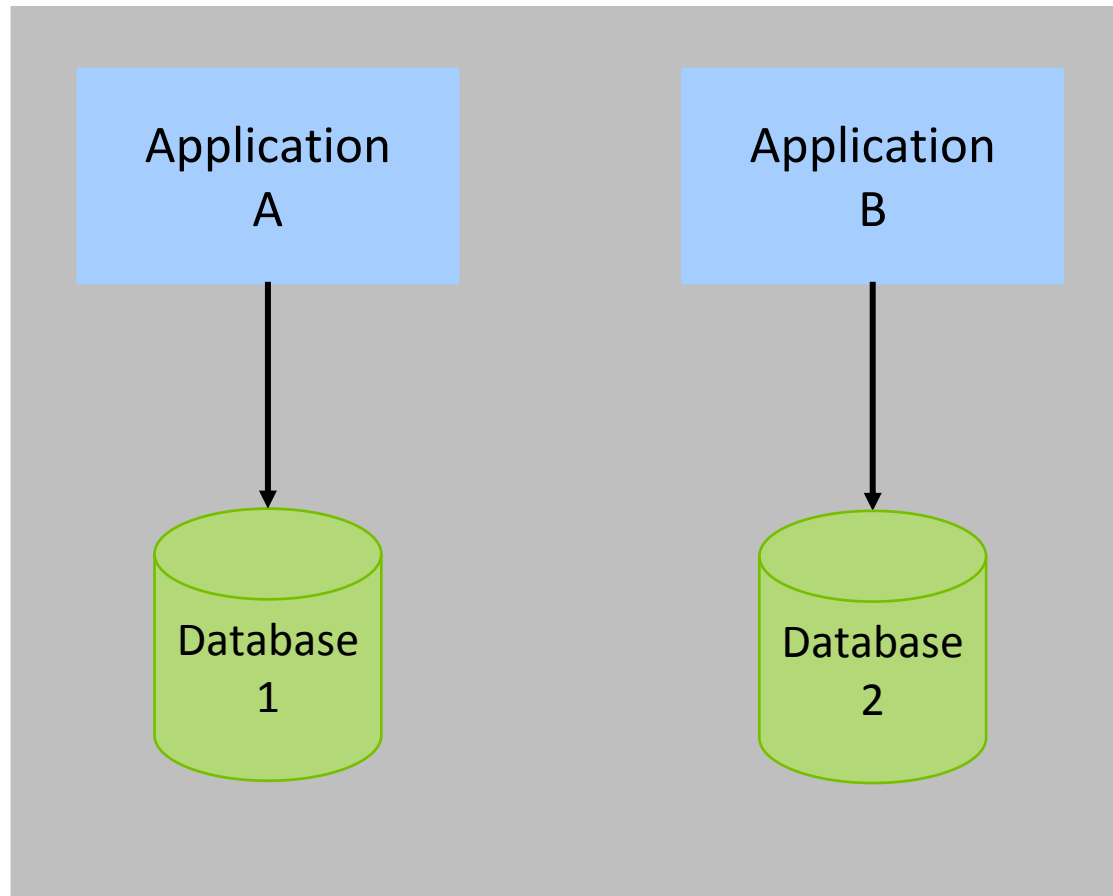
Integration databases have been used for years



Polyglot persistence doesn't work here



Application databases do not share it's data

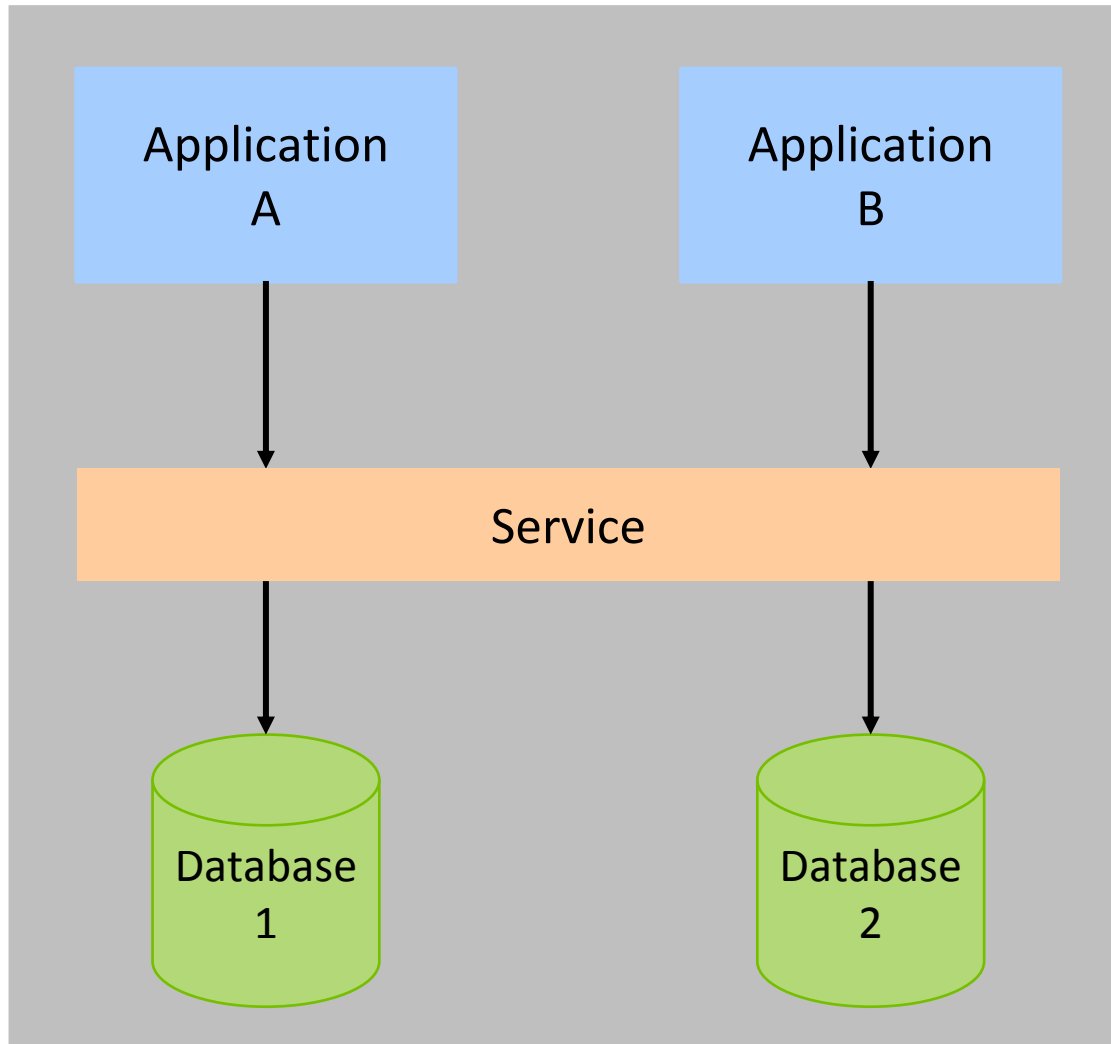




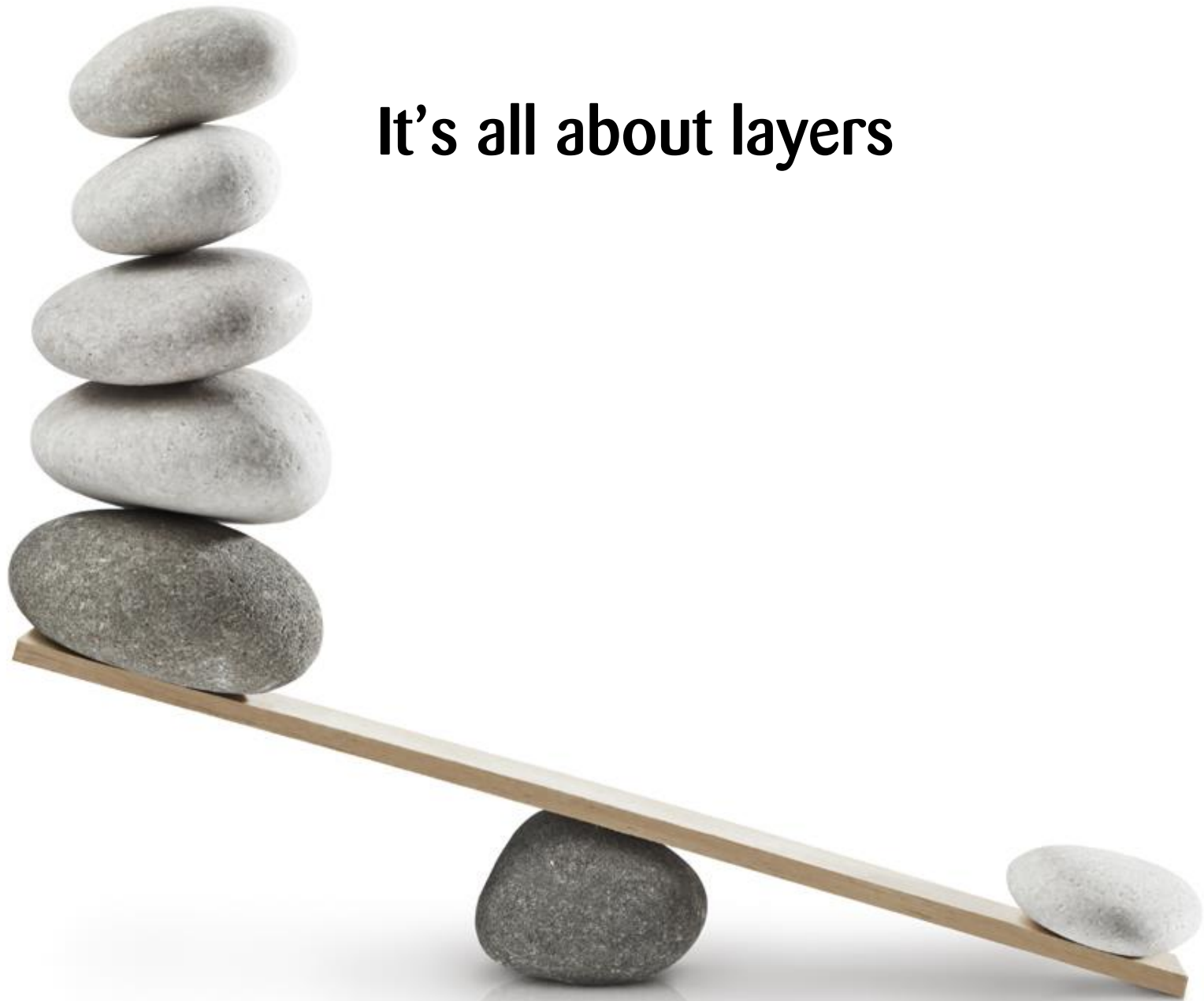
Borat @BoratNoSQL

Fine! But I have not only one application.

Application database with SOA



It's all about layers



Well known layers

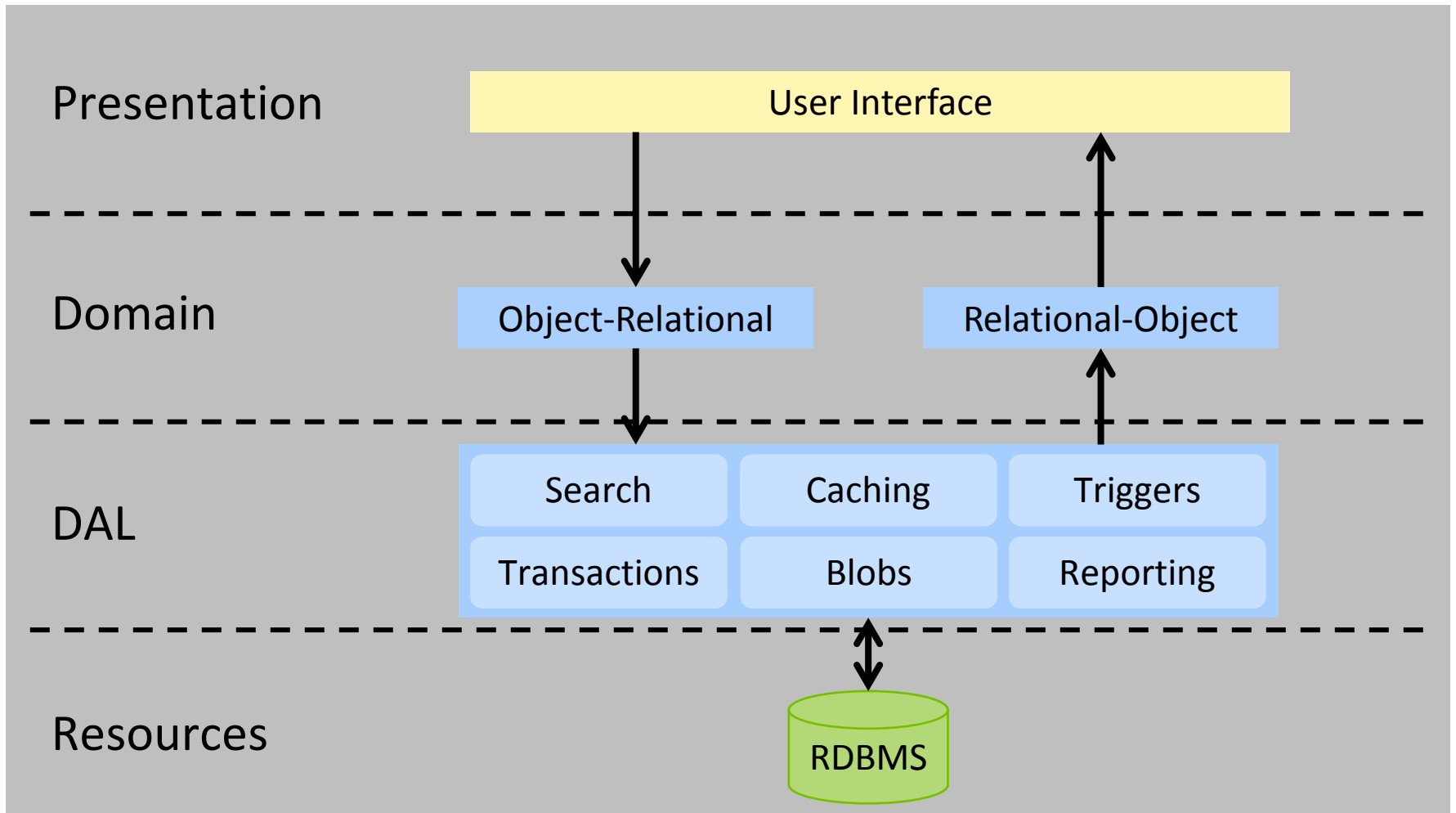
Presentation Layer

Domain Layer

Resource Access Layer (Data Access Layer)

Resources

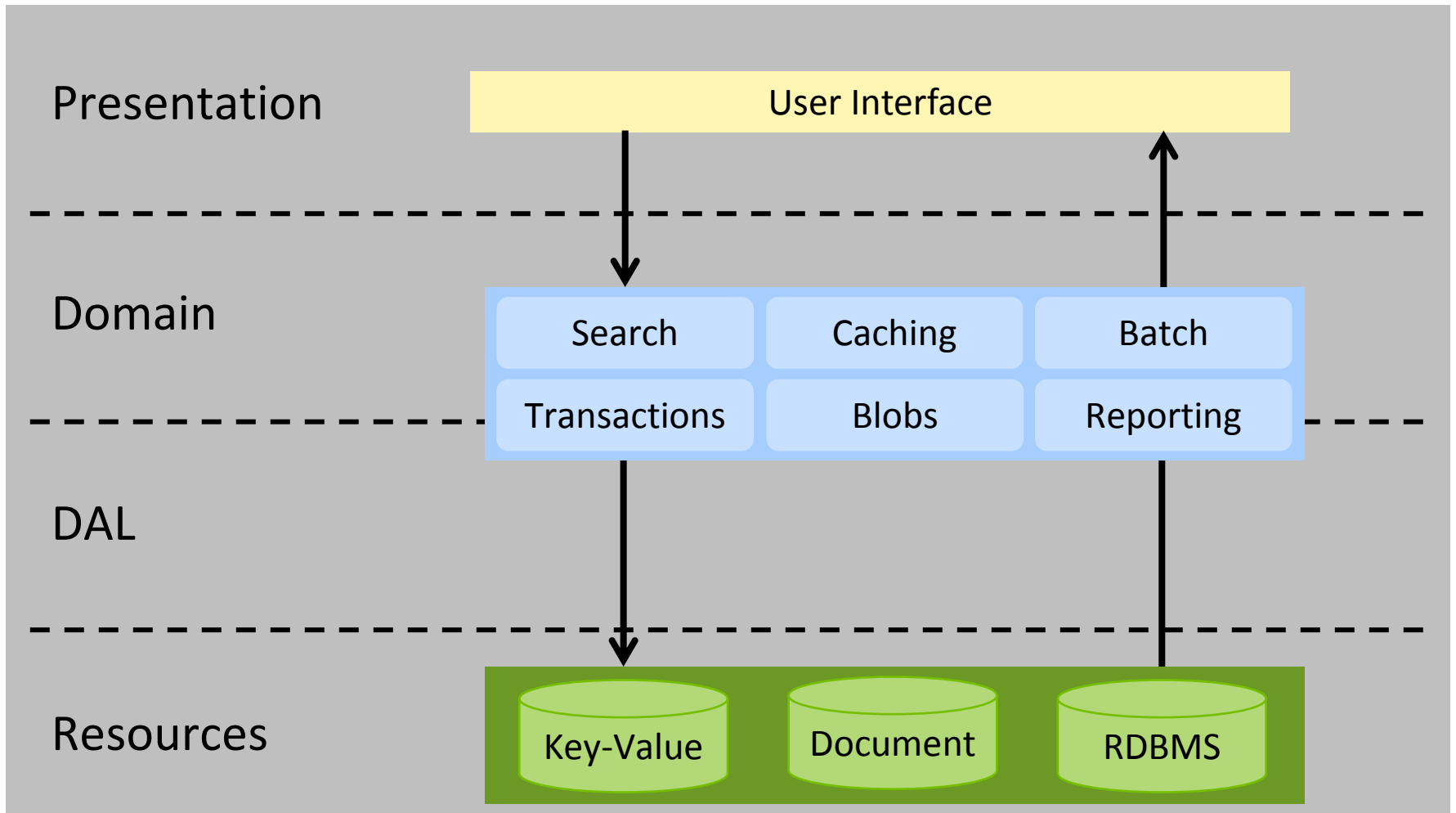
Common data tier design



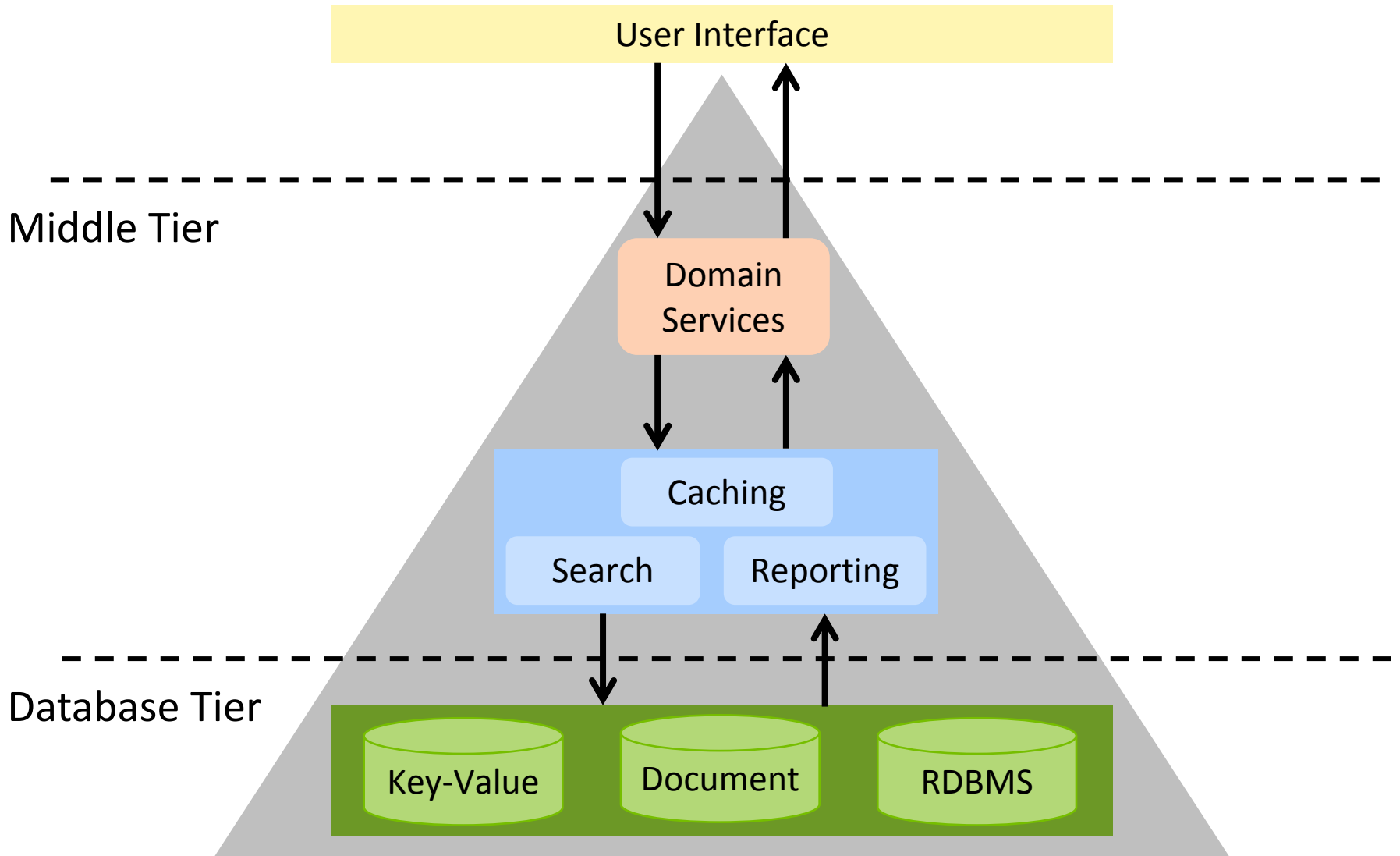
The truth of reusability



Data access with reusable and seamless services



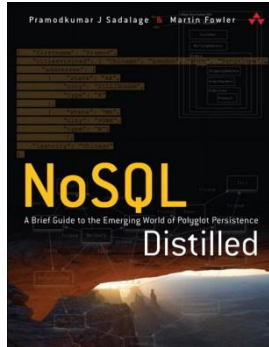
Putting all together



Use the right tool!



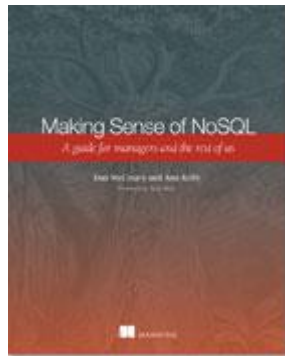
Resources



NoSQL Distilled

Author: Martin Fowler, Pramod J. Sadalage

ISBN: 978-0321826626



Making Sense of NoSQL

Author: Dan McCreary, Ann Kelly

ISBN: 978-1617291074

Links

<http://nosql-database.org/>

<http://en.wikipedia.org/wiki/NoSQL>

Thank you!

