Architectural Complexity

Lessons from the bwin P5 Poker System

Presented by:
Henrik "Henke" Lagercrantz & Gerold "Cactus" Kathan
QCon London 2010, March 10 2010

Online Poker

- Functional Requirements in the Poker domain are well-understood and therefore **EASY** to implement
- **However,** there are a series of Non-Functional requirements that complicate the implementation of Online Poker

extreme performance

massive stability

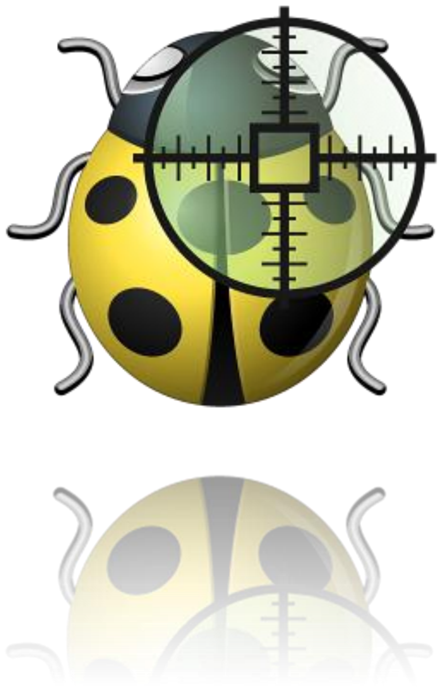
immediate time2market

infinite flexibility

seamless integration

maximum cost efficiency

superior quality



cosy customer experience

customer care friendly

tons of real money

real-time transactions

ultimate security

fraud detection

auditable compliance

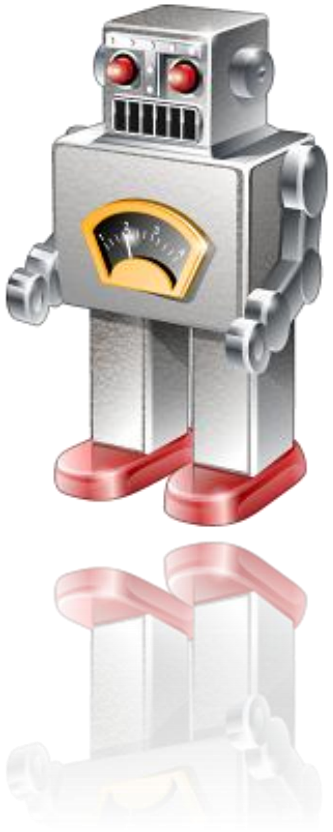
global reach

organizational scaling

operational excellence

pervasive monitoring

fully automated processes



hold on

every serious

e-commerce

shop has to deal with those issues.....

BUT...

B2B

SaaS / fully managed service

multi tenancy

white labeling

fully customizable clients

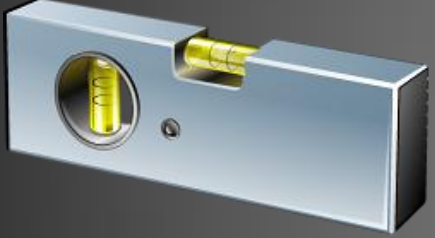
B2G

weird regulations

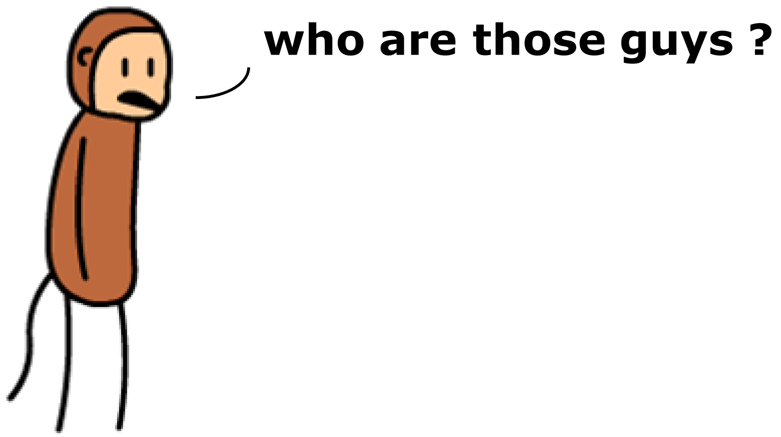
local installability

identity control

self-learning fraud analysis



responsible gaming



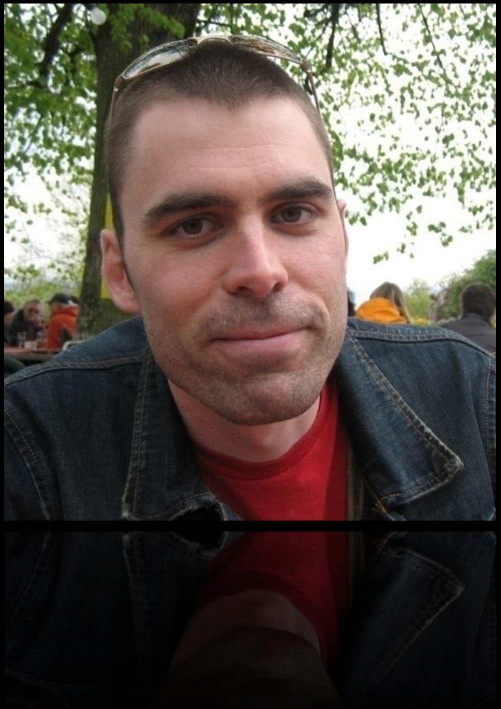
...getting

COMPLEX ?

The Speakers

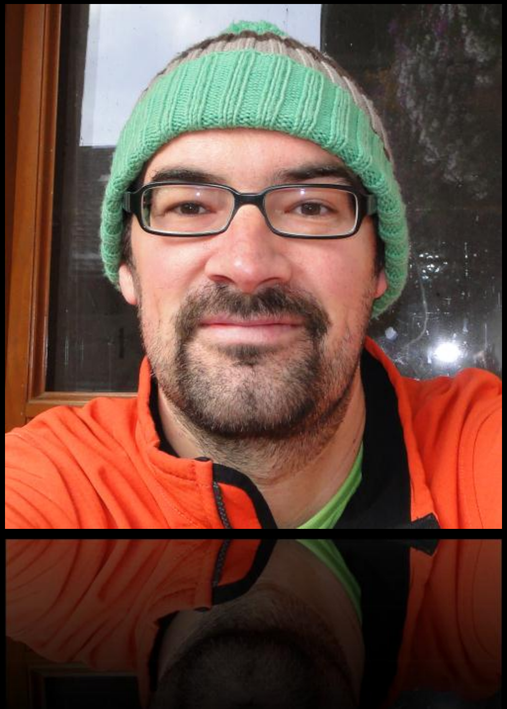
Henke

“I would prefer the correct solution over the simple one"



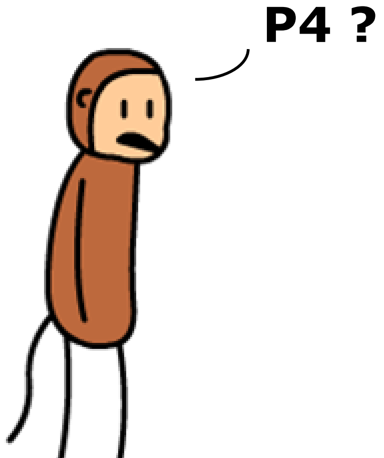
Cactus

"...guys, what is the real problem you want to solve here ?"



bwin

P5 HISTORY



What was P4?

- P4 = Poker v4
- Ran from 2002 to 2008...it worked!

  But...we had performance issues, that we fixed...

- Then the UIGEA happened...and we turned to Europe...

REGULATION!!

REGULATION

- Non-compliance = **NOT OPTIONAL!**
  - ‘Weird’ regulations/requirements

- Implementation on P4

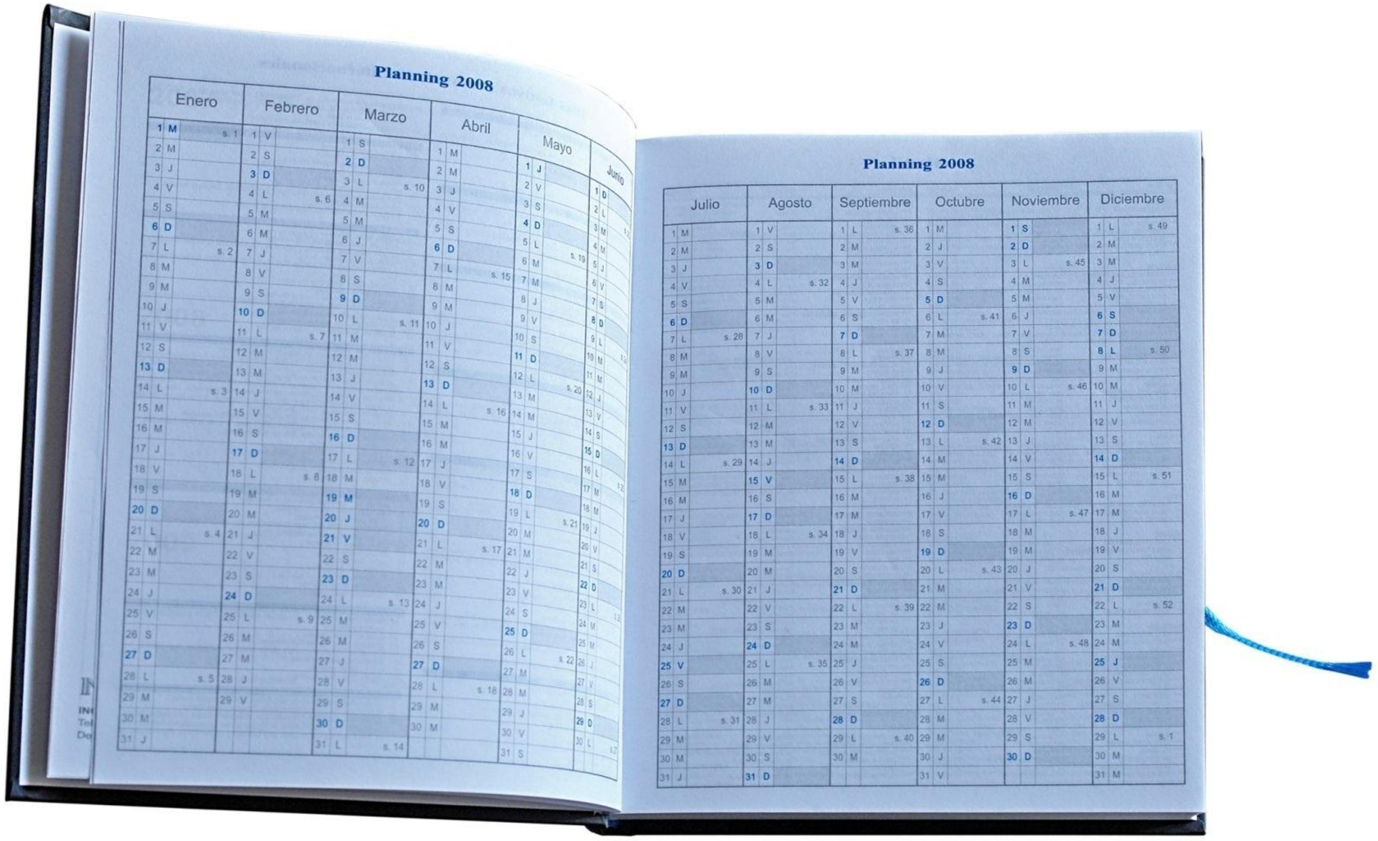
= **IMPRACTICAL** for multiple markets

ONE WAY

NEW Build

= P5

bwin



2007 challenging project ahead...

business drivers

Support the widest range of gaming

regulatory requirements

Build a poker platform with a **highly customizable** client.

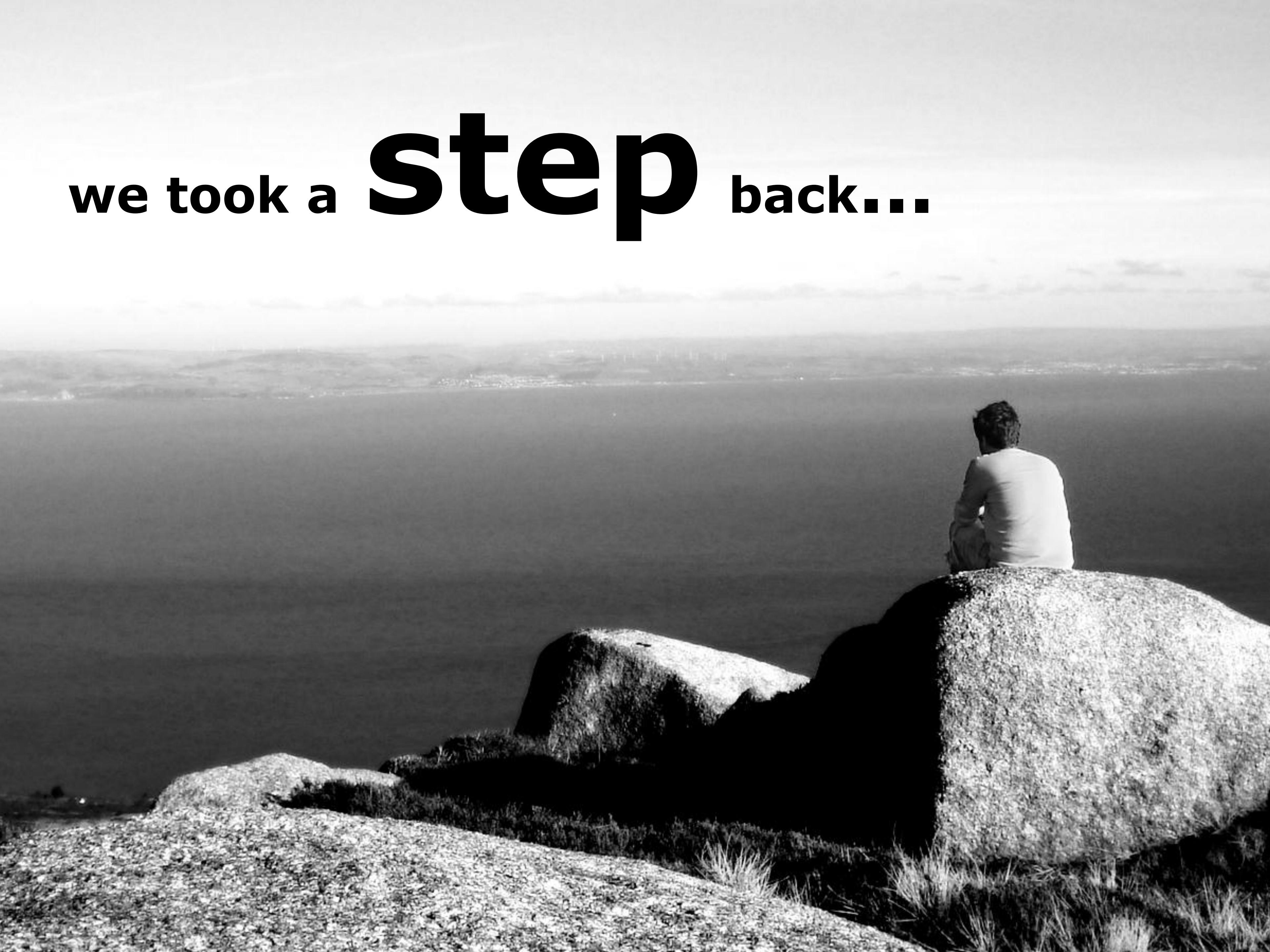
Enable a **wide range**

of new games, business models, and integration scenarios

bwin

Build the world's **most scalable** and **cost-efficient** poker platform.

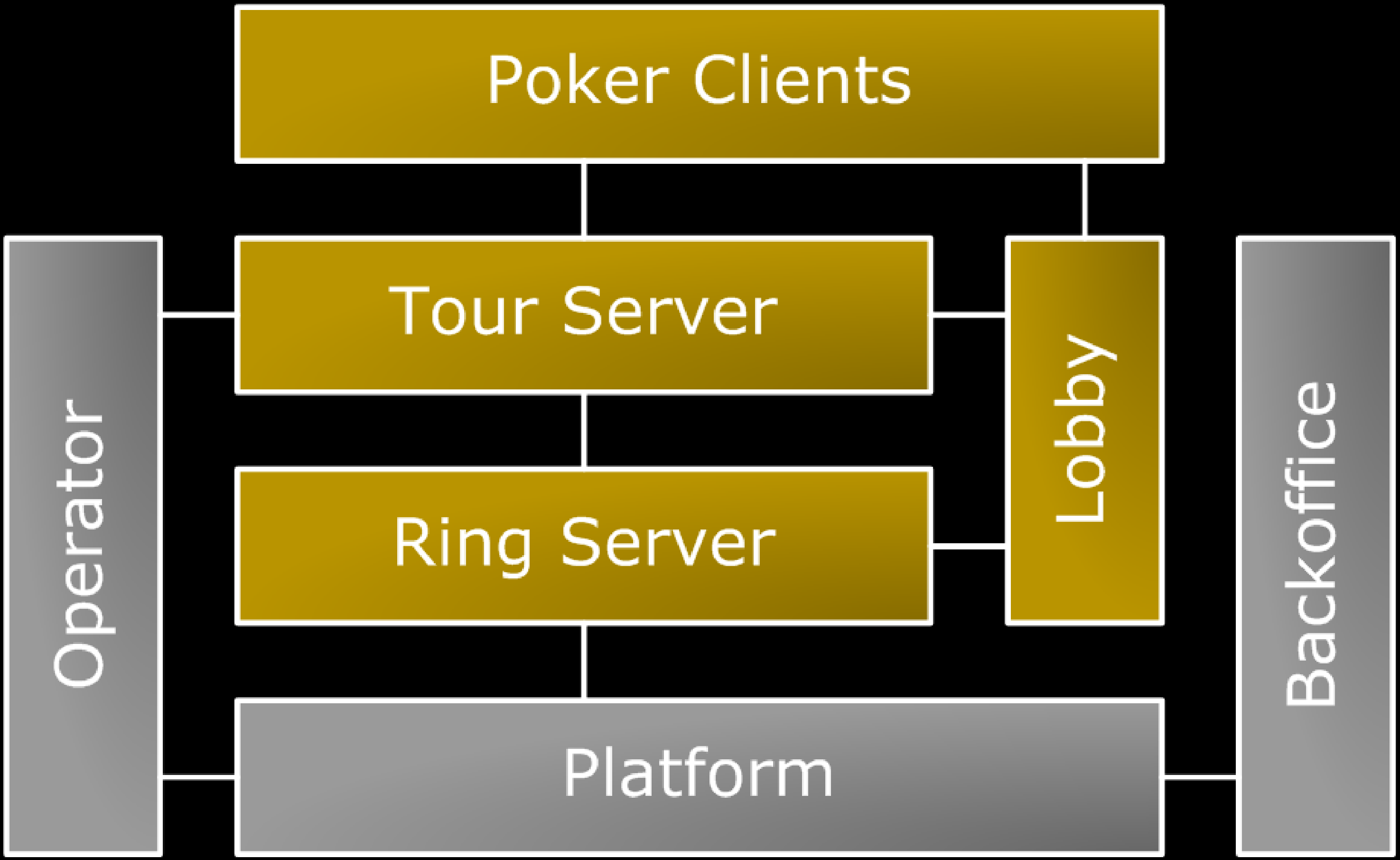
we took a **step** back...



we could **feel** the complexity of P4 …

But it looks simple?

bwin



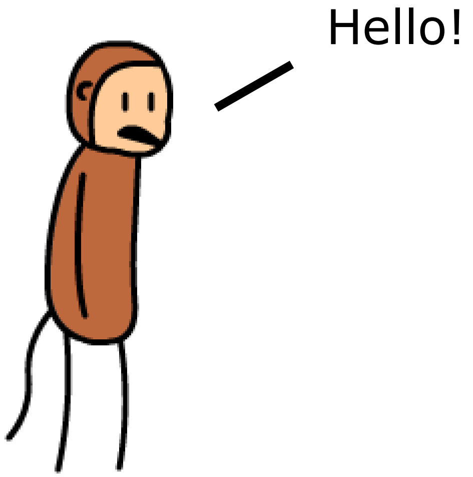
bwin

we looked **inside** the simple boxes …

Buuuuh .... good old legacy **mess** we learned to

differentiate…

Accidental Complexity



```
    public synchronized void acceptInput(String s, int seat)
    {
       inStrings[qB]=s;
       inSeats[qB]=seat;
       qB=(qB+1)&127;
//     interrupt();
    }

    public void processInput(String s, int seat)
    {
       now=System.currentTimeMillis();
       myRoom.rnd.setSeed(now);
       switch(s.charAt(0))
       {
          case 56: // Chat text
             pk=s.length();
             char tc1[]=new char[pk+1];
             tc1[0]=(char)56;
             tc1[1]=(char)seat;
             s.getChars(1,pk,tc1,2);
             pi=2;
             pj=2;
             while (pi<=pk) // Trim multiple spaces
             {
                if (tc1[pi]==':')
                   tc1[pi]=' ';
                if (tc1[pi]!=' ' || tc1[pj-1]!=' ')
                   tc1[pj++]=tc1[pi];
                pi++;
             }
             sqlTier.logChat(seated[seat].ID,createdNum,hand,s.substring(1));
             addMessage(new String(tc1,0,pj));
             break;
          case 57: // Add this much cash
//           pj=sqlTier.reserveMoney(seated[seat],seated[seat].cash+decodeInt(s,1)
             sqlTier.reserveMoney(seated[seat],seated[seat].cash+decodeInt(s,1),th
             char tc2[]=new char[9];
             tc2[0]=(char)57;
             tc2[1]=(char)seat;
             pi=2+encodeInt(seated[seat].cash,tc2,2);
             addMessage(new String(tc2,0,pi));
             tc2[0]=(char)58;
             pi=1+encodeInt(pj,tc2,1);
             seated[seat].insertPrivateMessage(new String(tc2,0,pi));
             break;
          case 59: // Request to sit in or out
//           System.out.println("Sit in/out: "+(int)s.charAt(1));
             char tc3[]=new char[3];
             tc3[0]=(char)59;
             tc3[1]=(char)seat;
```

Essential

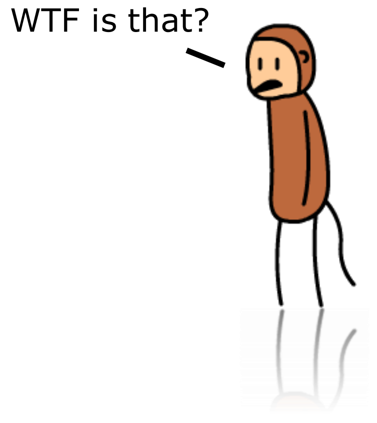
Complexity

$$E = mc^2$$

"In the presence of essential complexity, establishing simplicity in one part of a system requires trading off complexity in another"

– Grady Booch, IT Guru

bwin



P5 Architecture

KEY Principles

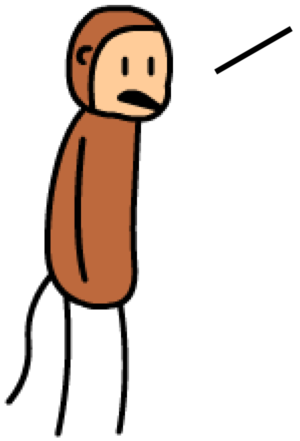
- Modularisation
- Asynchronous Integration
- Abstraction
- Encapsulation

Eventual Consistency

bwin

"Those are my principles, and if you don't like them... well, I have others"

- Groucho Marx

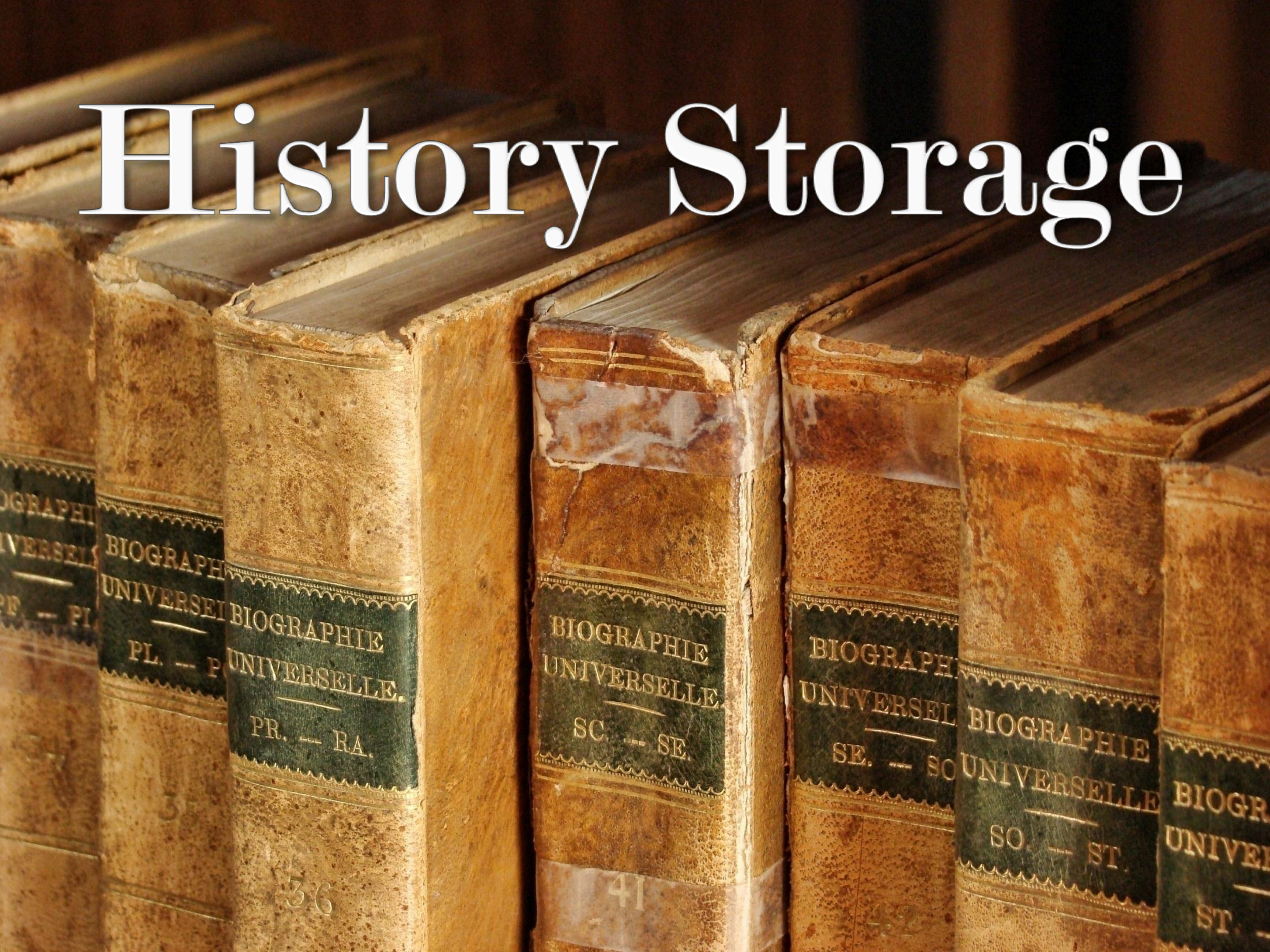

too much fluffy stuff already...
details pleeese!

bwin

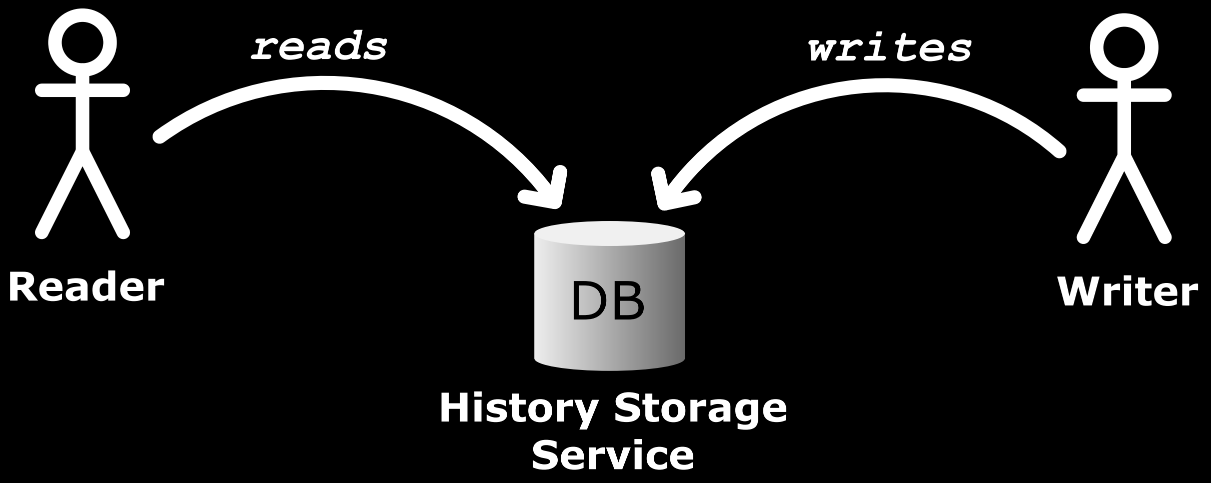
Sorry about that ...

...let's look at an **Example** instead

History Storage



Original Setup



Writing

bwin

Availability

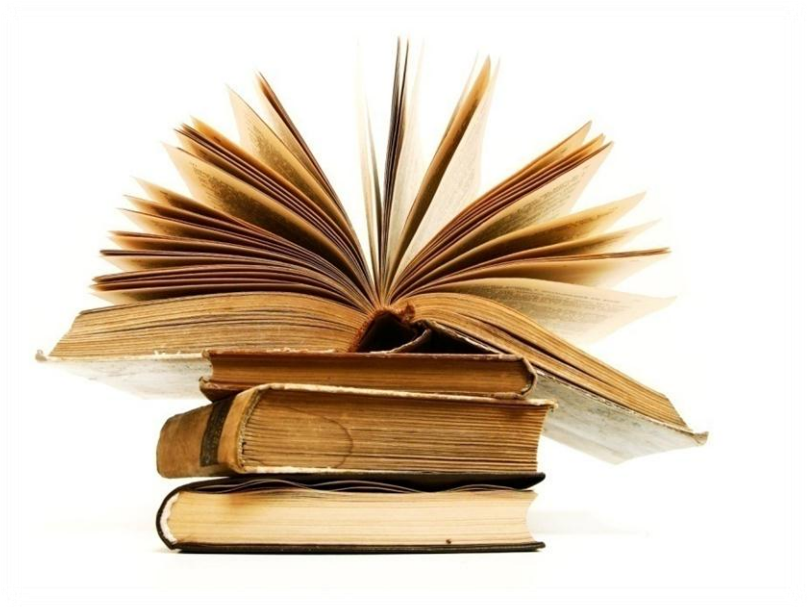
Performance

Flexibility

Consistency

bwin

Reading

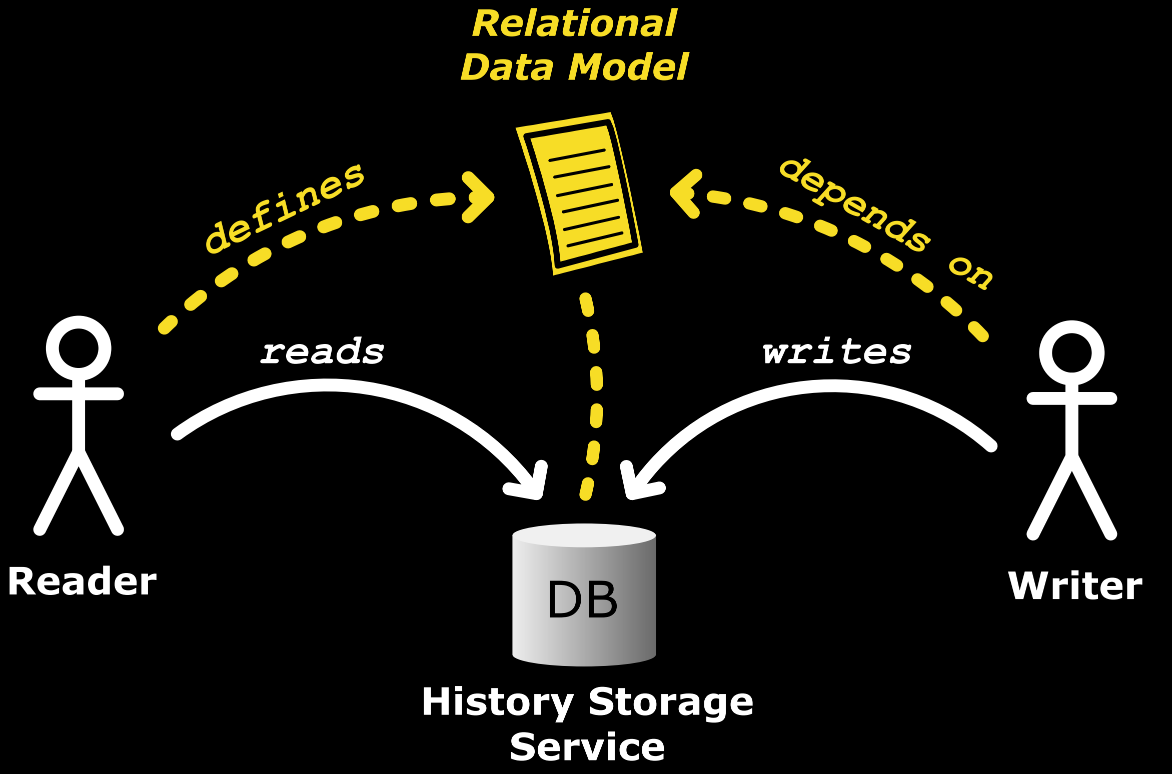


Availability

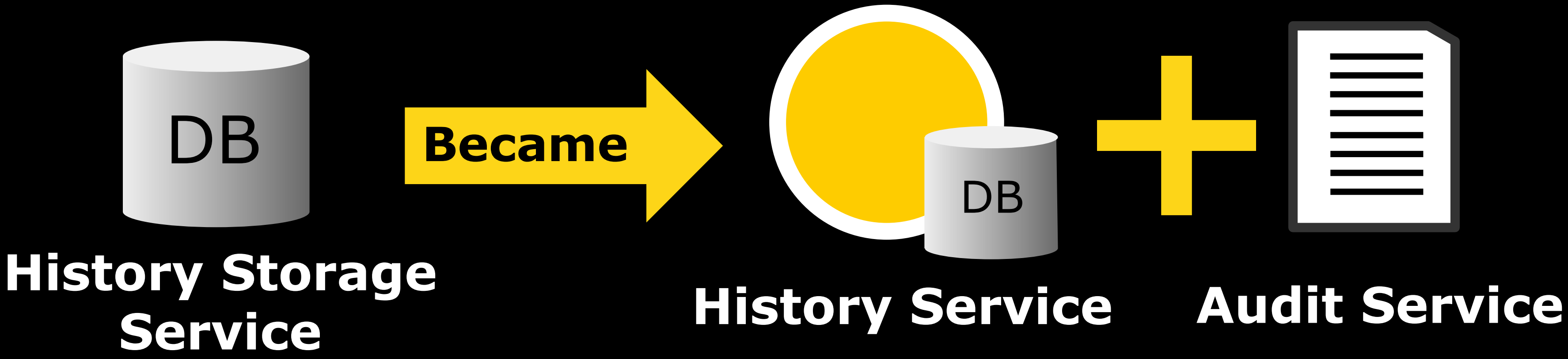
Performance

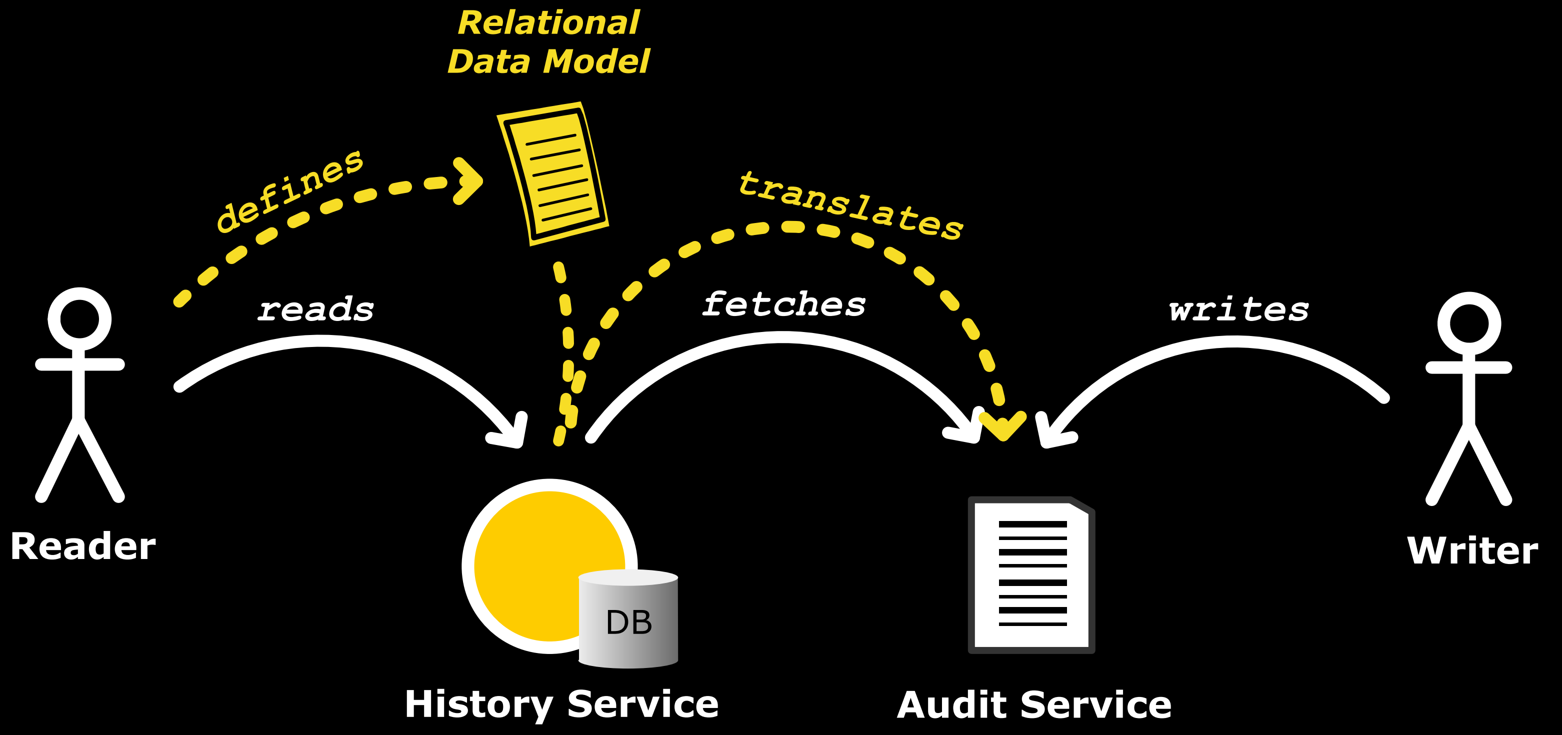
Flexibility

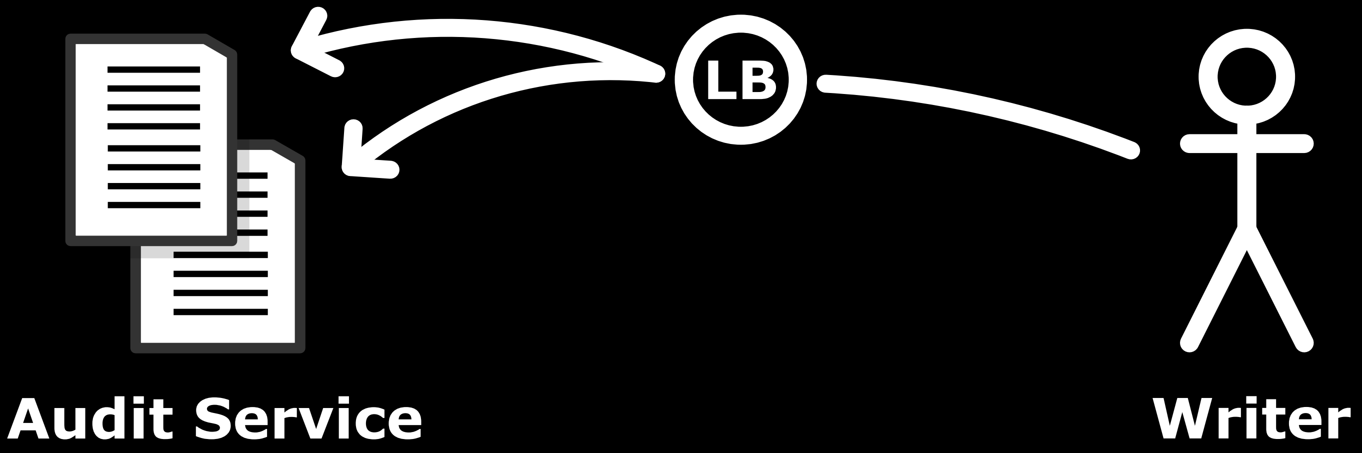
Consistency


Relational Data Model
defines
depends on
reads
writes
Reader
Writer
DB
History Storage Service

Time to Modularize...

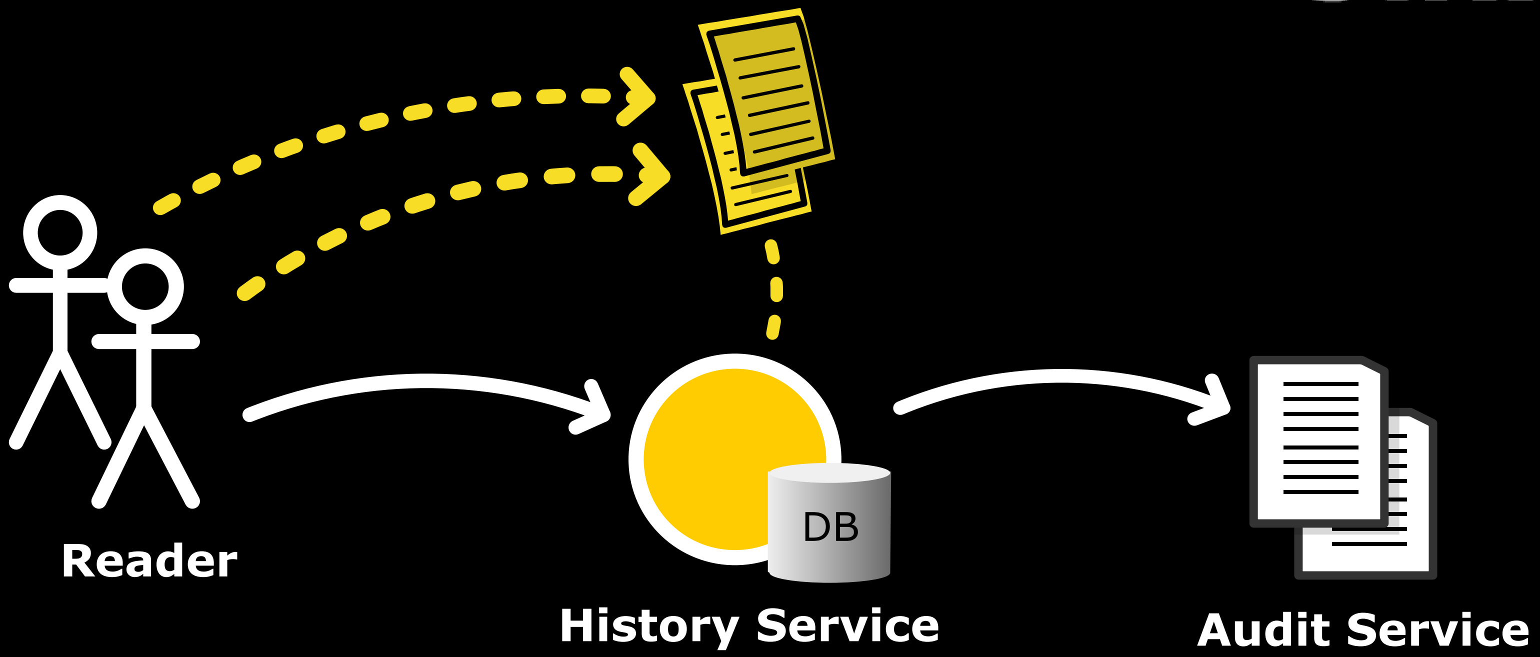



Relational
Data Model
defines
translates
reads
fetches
writes
Reader
Writer
DB
History Service
Audit Service



High Performance

Continuous Availability



Consistency

Flexibility

Busting a CAP (theorem)

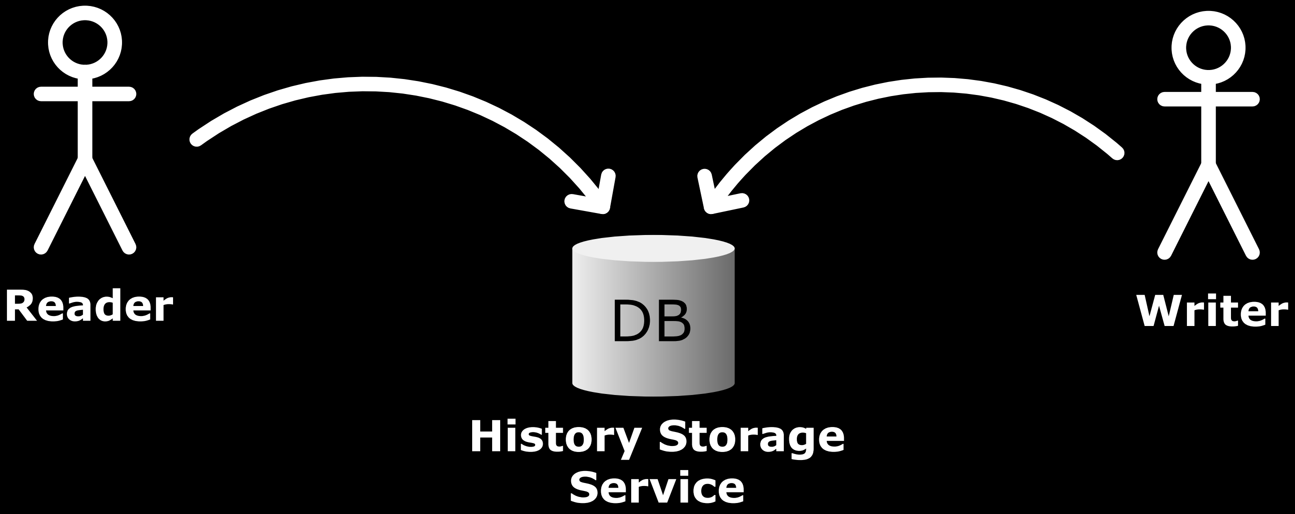
High Performance

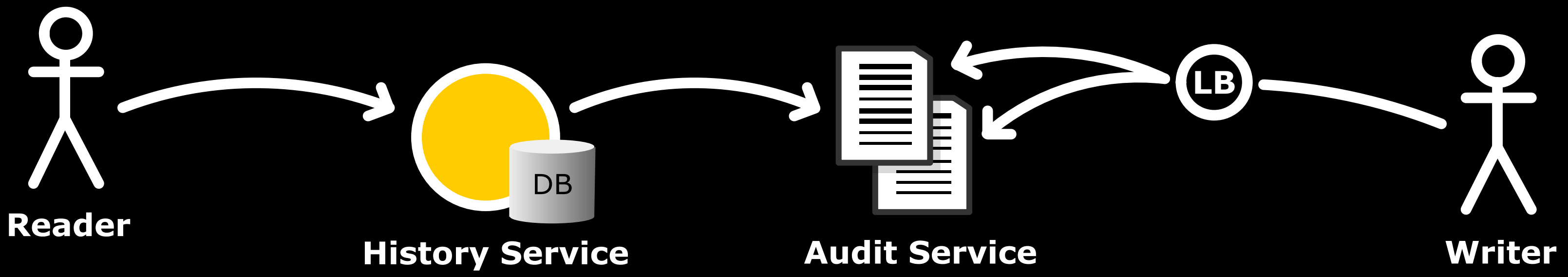
Continuous Availability

Flexibility

~~*Consistency*~~

Eventual Consistency


Reader
DB
Writer
History Storage Service


Reader
History Service
DB
Audit Service
LB
Writer

fine ...

short

recap...

bwin

we did a **big** rewrite of our

poker system

we **won't** do it again ;-)

we **learned** our lessons !

bwin

do not try to escape

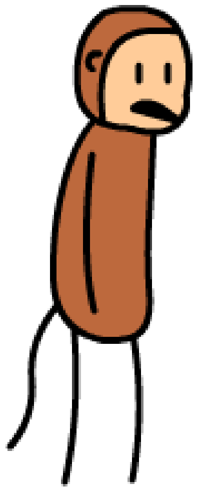
the **essential** complexity !

architect it !

bwin

resistance is **futile**

bwin



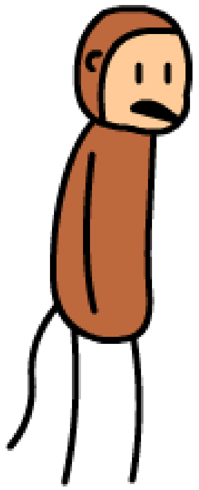
any questions ?

you remember the cat ...

bwin

SIR, CATS CAN'T LIVE ON LASAGNA.
BUT I READ THIS COMIC-
CATS CAN'T LIVE ON LASAGNA!
HAHA, LOOK HOW LAZY HE IS!!
NO. HE'S DEAD. YOU KILLED YOUR CAT.
Cyanide and Happiness © Explosm.net

join.us@bwin.com



Embrace Essentíal Complexity.
Architect it !