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 wellunderstood 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



B2**B**



SaaS / fully managed service mix wswaeg service

multi tenancy

white labeling



fully customizable clients



weird regulations





local installability



identity control

self-learning fraud analysis USng SUS[A2]2







Leshousinie Aguinia





who are those guys ?



The Speakers

Henke

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





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






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





NEW Build = P5

bwin

	Enero	Fe	ebrero	Marzo		_								
	1 M	s. 1 1 V			Abril		_)						
-10	2 M	2.8	1 5	and a shall be	1 M	Mayo								
	3 J	3 D	2 D		2 M	1 1	Jun			Plann	ing 2008			
	4 V	41	3 L	8.10	L	2 V	10							1
10	5 S	5 14	5.5 4 M		V	35	2	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciel	mbre
	6 D	E M	5 M	-	S	4 D	30 1	- Ind	1 V	1 L 5.3	6 1 M	1 5	1 L	s. 49
	7 1	2 7 1	6 J	6	D	5 L	44	2 M	2 5	2 M	2 1	2 D	2 M	
t	8 M	- 1 3	7 V	7	1	6 M	19 5 1	31	3 D	3 M	3 V	3 L 5.45	3 M	
1	9 M	8 V	8 S	8	s. 15	7 M	6 y	4 V	4 L 5.	12 4 J	4 S	4 M	4 J	
1	0 1	9 5	9 D	0	11	8 J	10	5 5	5 M	5 V	5 D	5 M	5 V	
		10 D	10 L	\$ 11 10	1M	9 V	8 0	6 D	6 M	6 S	6 L 8.4	1 6 J	6 S	
	1 1	11 L	s.7 11 M		3	10 S	9	7 4 5.	28 7 J	7 D	7 M	7 V	7 D	1000
1	2 5	12 M	12 M		V	11 D	10 M L	8 M	8 V	8 L	7 8 M	8 S	8 L	s. 50
1	3 D	13 M	13 1	12	S	12 1	It u	9 M	9 S	9 M	9 1	9 D	9 M	
14	4 L 5.	3 14 J	14 V	13	D	13 M	20 12	10 J	10 D	10 M	10 V	10 L s. 4	i 10 M	
15	5 M	15 V	15 0	14	L s. 16	14 14	13 0	11 V	11 L 5.1	13 11 J	11 S	11 M	11 J	
16	M	16 5	10 5	15	M	5	14 0	12 S	12 M	12 V	12 D	12 M	12 V	
17	J	17 0	16 D	16	M	6 11	15 0	13 D	13 M	13 S	13 L 5.4	2 13 J	13 S	
18	V	10	17 L	\$ 12 17	1		16	14 L 5.3	29 14 J	14 D	14 M	14 V	14 D	A DELINE AND A
10	c	10 L	5.8 18 M	18	V	15	17 . 17	15 M	15 V	15 L \$.3	8 15 M	15 S	15 L	s. 51
20	0	19 M	19 M	19	S	80	10	16 M	16 S	16 M	16 J	16 D	16 M	
20	U	20 M	20 J	20	D	9 L 8.2	1 10	17 J	17 D	17 M	17 V	17 L 5.4	7 17 M	
21	L 5.4	21 J	21 V	21	2	M	12	18 V	18 L S.	4 18 J	18 S	18 M	18 J	
	M	22 V	22 S	22	5. 17 2	M	au V	19 S	19 M	19 V	19 D	19 M	19 V	CRIEDING ST
	M	23 S	23 D	22	2	2 3	15	20 D	20 M	20 S	20 L s.4	3 20 J	20 5	
24		24 D	24 1	# 12 00	23	V	40	21 L 5.3	0 21 J	21 D	21 M	21 V	21 D	
25	V		8 9 25 14	a. 13 24 .	24	S	41 1	22 M	22 V	22 L 5.3	9 22 M	22 5	22 L	\$.52
26		26 M	3 0 20 nl	25 \	25	D	24 M	23 M	23 S	23 M	23 J	23 D	23 M	
27 1	0	20 14	26 M	26 S	26	1	25 M	24 J	24 D	24 M	24 V	24 L 5.4	8 24 M	
		21 M	27 3	27 0	27	M 1.0		25 V	25 L 8.3	5 25 J	25 S	25 M	25 J	
	5.5	28 J	28 V	28 L	5 18 29	11	27 V	26 S	26 M	26 V	26 D	26 M	26 V	The sea
29 N		29 V	29 S	29 M	20	-	28 \$	27 D	27 M	27 S	27 L s.4	4 27 J	27 5	
30 N	1		30 D	30 M	29	-	29 D	28 L 8.3	1 28 J	28 D	28 M	28 V	28 D	
1 1			31 1	e 14	30	V		29 M	29 V	29 L 5.4	0 29 M	29 S	29 L	s. 1
1000		-		a. 14	31	S		30 M	30 S	30 M	30 J	30 D	30 M	And the second second

A COMPANY

2007 challenging project ahead...



business drivers



Support the widest range of gaming

requirements



Build a poker platform with a highly Customizable

client.



Enable a Wide range

of new games, business models, and integration scenarios



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?





we looked **INSICE** the simple boxes ...

Buuuuh good old legacy



we learned to differentiate...

public synchronized void acceptInput(String s, int seat)

```
inStrings[qB]=s;
inSeats[qB]=seat;
qB=(qB+1)$127;
interrupt();
```



Accidental Complexity



```
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

```
11
```

```
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;
```

```
11
```



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







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

bwin



too much fluffy stuff already... details pleeese!







Sorry about that ...

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

History Storage

BIOGRAPHIE

UNIVERSELLE.

SC -- SE

minu

BIOGRAPH

UNIVERSEL BIOGRAPHIE

SE. -- SOUNIVERSELLE BIOGR

SO. - ST

UNIVE

ST. -

GRAPHT

Million Contraction

VERSELS BIOGRAPH

UNIVERSETBIOGRAPHIE

PL. - PUNIVERSELLE.

PR. -- RA.



Original Setup





Writing

Availability

Performance

Flexibility

Consistency



Reading



Availability

Performance



Consistency







Time to Modularize...












High Performance Continuous Availability





Consistency

Flexibility



Busting a CAP (theorem)

High Performance



Continuous Availability

Flexibility

Consistency

Eventual Consistency











fine ...



short recap



we did a big rewrite of our poker system



we WON't do it again ;-)



we learned our lessons !



do not try to escape





architect it!



resistance is futile







you remember the cat ...





join.us@bwin.com





Embrace Essentíal Complexity. Architect it !