The pitfalls of inconsistency in modern systems

Jan Pustelnik

@gosubpl

Jonas Boner *(inspired by Pat Helland)* *http://jonasboner.com/talks/ - Life beyond (the) Illusion of Present / Pat Helland – Life Beyond Distributed Transactions: an Apostate's Opinion / Pat Helland, Dave Campbell – Building on Quicksand (Memories, Guesses, Apologies)*

IF WE CAN'T COORDINATE AND CANNOT BE CERTAIN - IN REAL LIFE WE TAKE AN EDUCATED GUESS - AND WHEN WE ARE WRONG WE APOLOGISE

Are apologies always enough?

Tesla driver killed in crash with Autopilot active, NHTSA investigating

http://www.theverge.com/2016/6/30/12072408/tesla-autopilot-car-crash-death-autonomous-model-s

Apologies…

Enough?

Are apologies always enough?

IF WE CAN'T COORDINATE AND CANNOT BE CERTAIN - IN REAL LIFE WE TAKE AN EDUCATED GUESS - AND WHEN WE ARE WRONG WE GO TO JAIL

Sometimes people go to jail for what they do… this is also “Real life”, isn’t it?

“Four former bankers have been ***jailed*** *for their role in attempting to manipulate the interest rate benchmark Libor”*

What’s a long running transaction?

1. **Check Credit rating**
2. **Do some manual checks (2 days)**
3. **Decide whether to extend a loan or not**

3 steps ***Business Process***

Is this always consistent?

1. **Check Credit rating**
2. **Do some manual checks (2 days)**
3. **Decide whether to extend a loan or not**

What if ***Credit Rating*** changes while we are in ***Step 2***

We check the credit rating…

1. **Check Credit rating**
2. **Do some manual checks (2 days)**
3. **Decide whether to extend a loan or not**

What if ***Credit Rating*** changes while we are in ***Step 2***

Time passes…

1. **Check Credit rating**
2. **Do some manual checks (2 days)**
3. **Decide whether to extend a loan or not**

What if ***Credit Rating*** changes while we are in ***Step 2***

And our initial Guess was wrong…

1. **Check Credit rating**
2. **Do some manual checks (2 days)**
3. **Decide whether to extend a loan or not**

What if ***Credit Rating*** changes while we are in ***Step 2***

KA-BOOM

What do we do now?

1. **Check Credit rating**
2. **Do some manual checks (2 days)**
3. **Decide whether to extend a loan or not**

But Business people have already sorted this out – basing on initial Credit Check they promise you a loan, on condition that your rating does not change and then re-check

There are different sorts of “apologies”…

1. **Check Credit rating**
2. **Do some manual checks (2 days)**
3. **Decide whether to extend a loan or not**

Another Business Level decision might be to just go-ahead and acknowledge the risk from potential default.

But consequences may vary…

1. **Check Prohibited Country Status**
2. **Do some manual checks (2 days)**
3. **Decide whether to onboard or not**

So…

And we may not always be able to …

1. **Check Prohibited Country Status**
2. **Do some manual checks (2 days)**
3. **Decide whether to onboard or not**

LICENSE REVOKED

… apologise

1. **Check Prohibited Country Status**
2. **Do some manual checks (2 days)**
3. **Decide whether to onboard or not**

Wrong decision can have your license revoked – be careful…

You are not always free to just take the risk…

Is this problem visible in computer systems too? In distributed systems? In databases?

YES, we’ve known that for quite a long time...

Is this problem visible in computer systems too? In distributed systems? In databases?

YES, we’ve known that for quite a long time...

RFC 677 (1975) Maintenance of duplicate databases

Is this problem visible in computer systems too? In distributed systems? In databases?

YES, we've known that for quite a long time…

RFC 677 (1975) Maintenance of duplicate databases

Sagas Hector Garcia-Molina (1987) Long running transactions

Is this problem visible in computer systems too? In distributed systems? In databases?

YES, we've known that for quite a long time...

RFC 677 (1975) Maintenance of duplicate databases

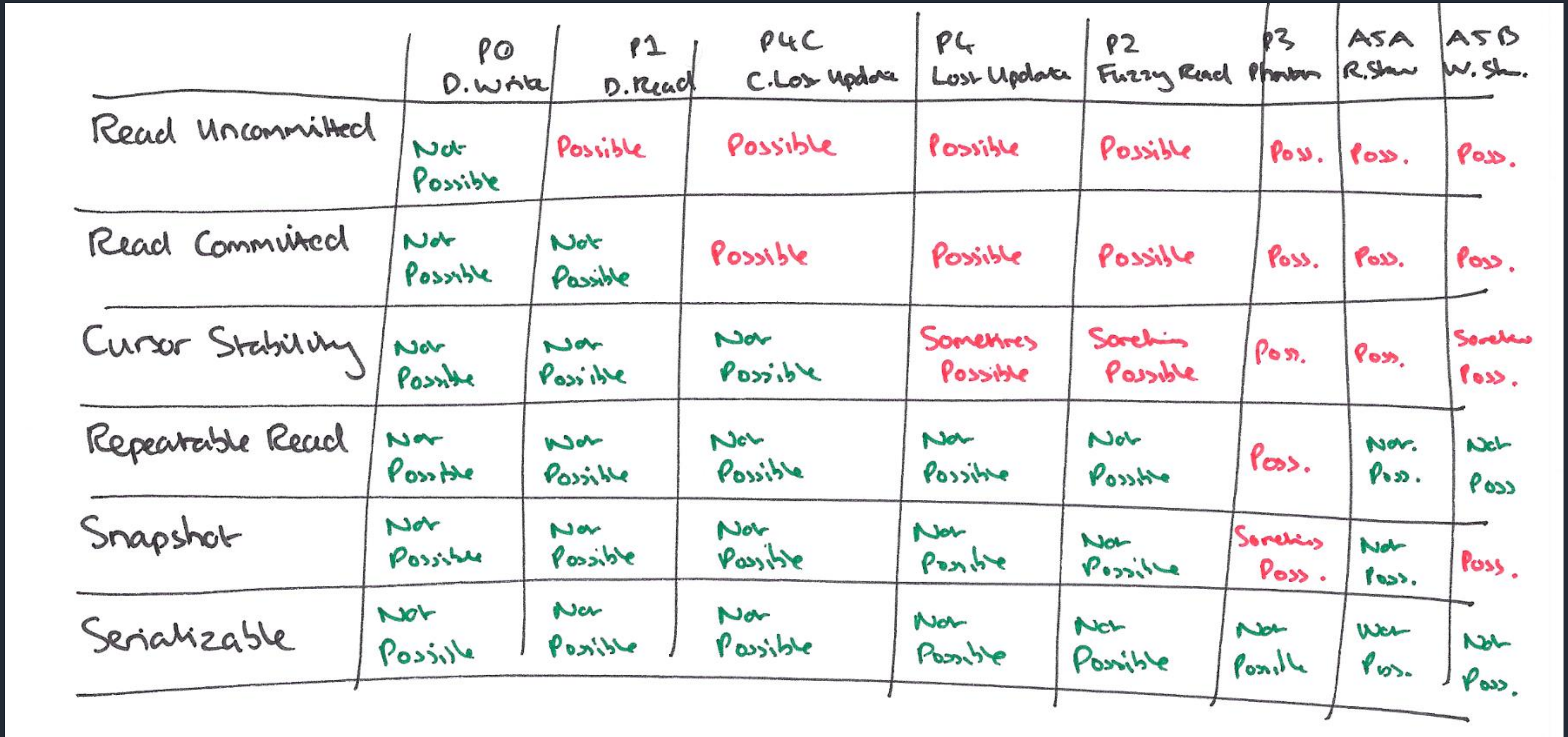
Sagas Hector Garcia-Molina (1987) Long running transactions

A Critique of ANSI SQL Isolation Levels

Berenson et al. (1995)

Let me show you how this works in a traditional database

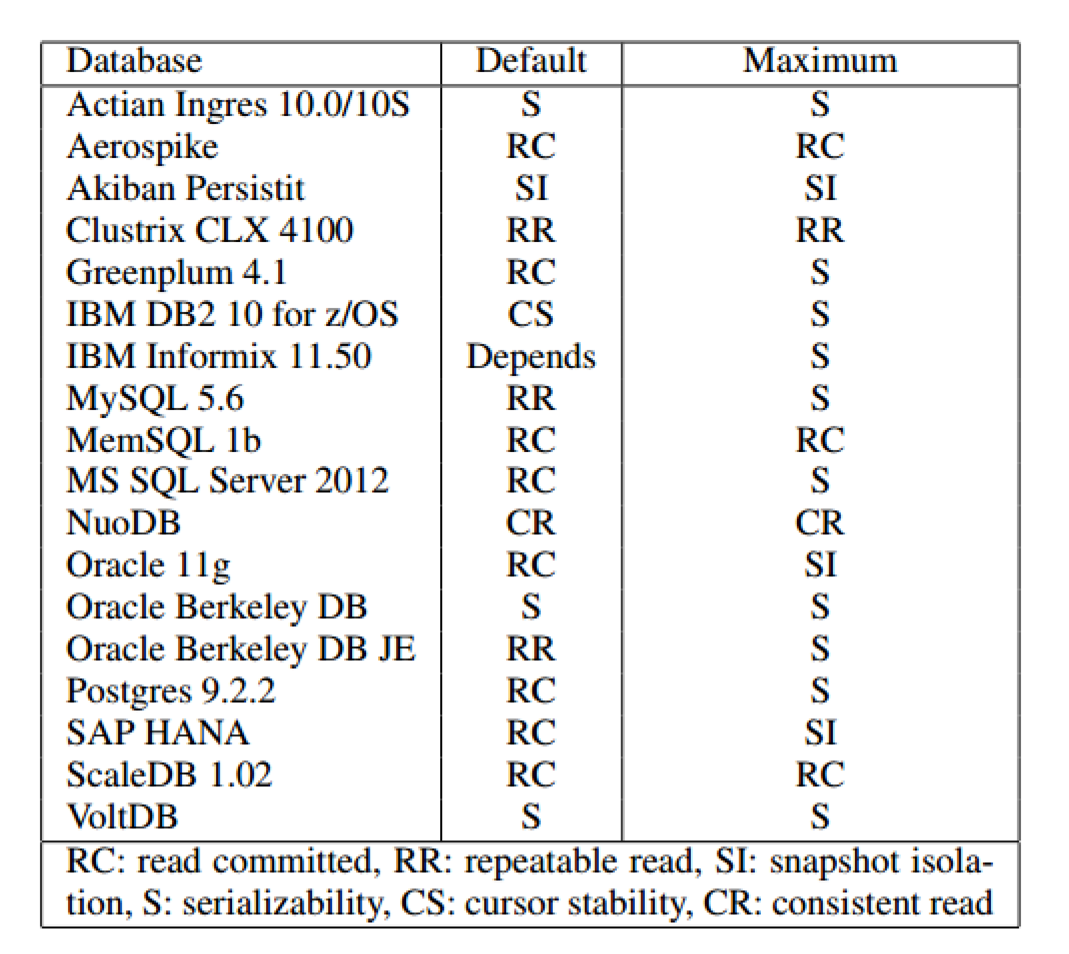
https://blog.acolyer.org/2016/02/24/a-critique-of-ansi-sql-isolation-levels/



	P0 D.Write	P1 D.Read	P4C C.Lost Update	P4 Lost Update	P2 Fuzzy Read	P3 Phantom	A5A R.Skew	A5B W.Sk.
Read Uncommitted	Not Possible	Possible	Possible	Possible	Possible	Poss.	Poss.	Poss.
Read Committed	Not Possible	Not Possible	Possible	Possible	Possible	Poss.	Poss.	Poss.
Cursor Stability	Not Possible	Not Possible	Not Possible	Sometimes Possible	Sometimes Possible	Poss.	Poss.	Sometimes Poss.
Repeatable Read	Not Possible	Not Possible	Not Possible	Not Possible	Not Possible	Poss.	Not. Poss.	Not Poss
Snapshot	Not Possible	Not Possible	Not Possible	Not Possible	Not Possible	Sometimes Poss.	Not Poss.	Poss.
Serializable	Not Possible	Not Possible	Not Possible	Not Possible	Not Possible	Not Possible	Not Poss.	Not Poss.

What are the common defaults? *Highly Available Transactions: Virtues and Limitations – Peter Bailis et al.*

http://www.bailis.org/papers/hat-vldb2014.pdf



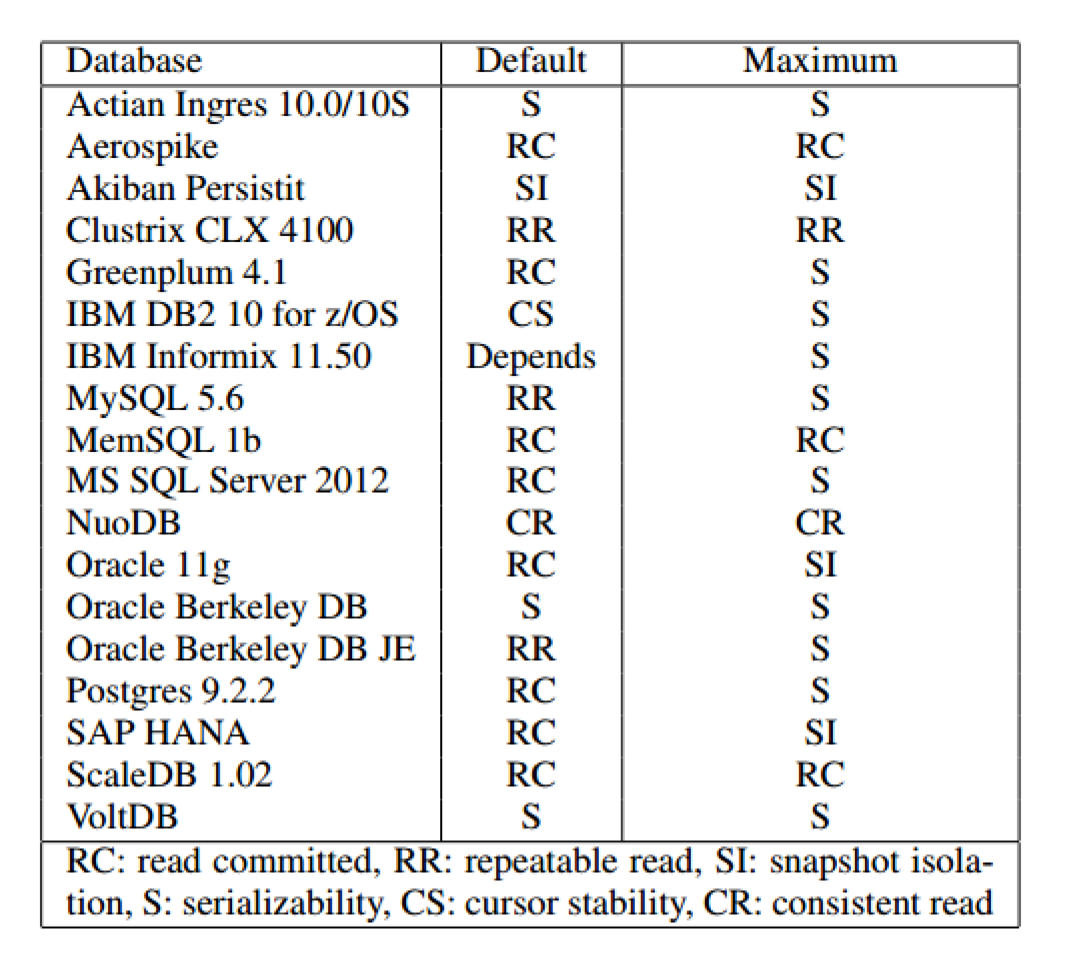
Database	Default	Maximum
Actian Ingres 10.0/10S	S	S
Aerospike	RC	RC
Akiban Persistit	SI	SI
Clustrix CLX 4100	RR	RR
Greenplum 4.1	RC	S
IBM DB2 10 for z/OS	CS	S
IBM Informix 11.50	Depends	S
MySQL 5.6	RR	S
MemSQL 1b	RC	RC
MS SQL Server 2012	RC	S
NuoDB	CR	CR
Oracle 11g	RC	SI
Oracle Berkeley DB	S	S
Oracle Berkeley DB JE	RR	S
Postgres 9.2.2	RC	S
SAP HANA	RC	SI
ScaleDB 1.02	RC	RC
VoltDB	S	S

RC: read committed, RR: repeatable read, SI: snapshot isolation, S: serializability, CS: cursor stability, CR: consistent read

What are the common defaults? *Highly Available Transactions: Virtues and Limitations – Peter Bailis et al.*

READ COMMITED – allows Lost Update – T1 reads x = 100, T2 reads x = 100, T1 finishes commiting and writes x = 110, T2 uses old *stale* value of x = 100, increases it by 30 and commits x = 130.

Why Serializable is not default??? – it impacts performance…



Database	Default	Maximum
Actian Ingres 10.0/10S	S	S
Aerospike	RC	RC
Akiban Persistit	SI	SI
Clustrix CLX 4100	RR	RR
Greenplum 4.1	RC	S
IBM DB2 10 for z/OS	CS	S
IBM Informix 11.50	Depends	S
MySQL 5.6	RR	S
MemSQL 1b	RC	RC
MS SQL Server 2012	RC	S
NuoDB	CR	CR
Oracle 11g	RC	SI
Oracle Berkeley DB	S	S
Oracle Berkeley DB JE	RR	S
Postgres 9.2.2	RC	S
SAP HANA	RC	SI
ScaleDB 1.02	RC	RC
VoltDB	S	S

RC: read committed, RR: repeatable read, SI: snapshot isolation, S: serializability, CS: cursor stability, CR: consistent read

Serialize all the things? *A Critique of ANSI SQL Isolation Levels, H. Berenson et al.*

Why SERIALIZABLE is not the default setting??? – (asked my Boss once… ☺) it severely impacts performance… and availability…

But is still an option…

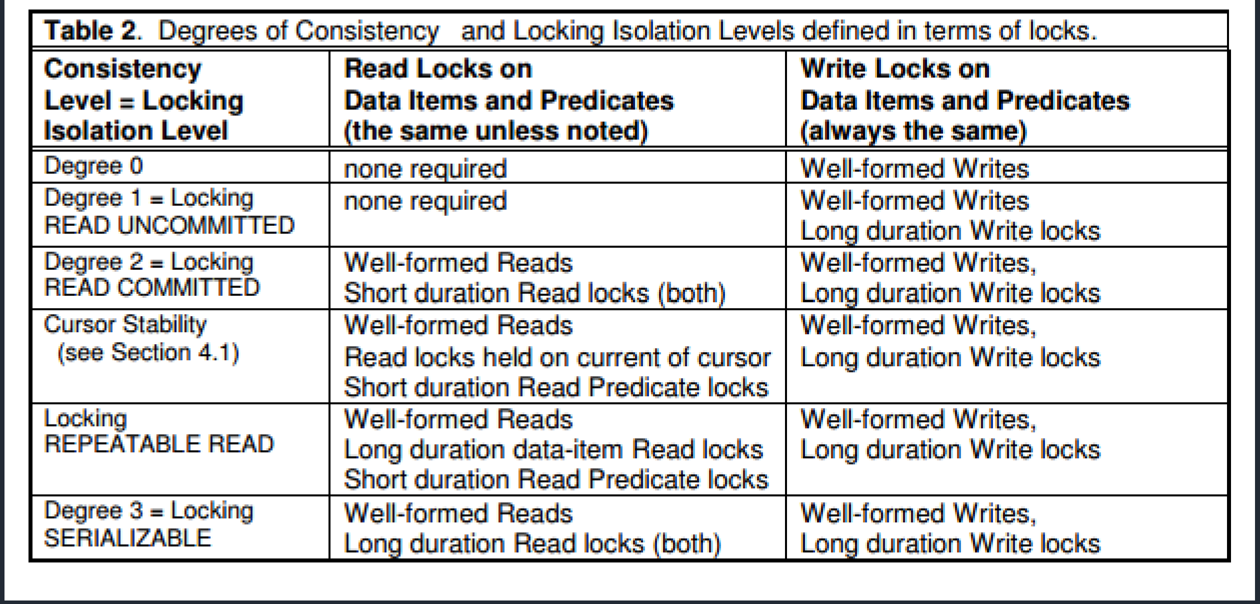


Table 2. Degrees of Consistency and Locking Isolation Levels defined in terms of locks.

Consistency Level = Locking Isolation Level	**Read Locks on Data Items and Predicates (the same unless noted)**	**Write Locks on Data Items and Predicates (always the same)**
Degree 0	none required	Well-formed Writes
Degree 1 = Locking READ UNCOMMITTED	none required	Well-formed Writes Long duration Write locks
Degree 2 = Locking READ COMMITTED	Well-formed Reads Short duration Read locks (both)	Well-formed Writes, Long duration Write locks
Cursor Stability (see Section 4.1)	Well-formed Reads Read locks held on current of cursor Short duration Read Predicate locks	Well-formed Writes, Long duration Write locks
Locking REPEATABLE READ	Well-formed Reads Long duration data-item Read locks Short duration Read Predicate locks	Well-formed Writes, Long duration Write locks
Degree 3 = Locking SERIALIZABLE	Well-formed Reads Long duration Read locks (both)	Well-formed Writes, Long duration Write locks

So why didn’t all of us heard about this problem already?

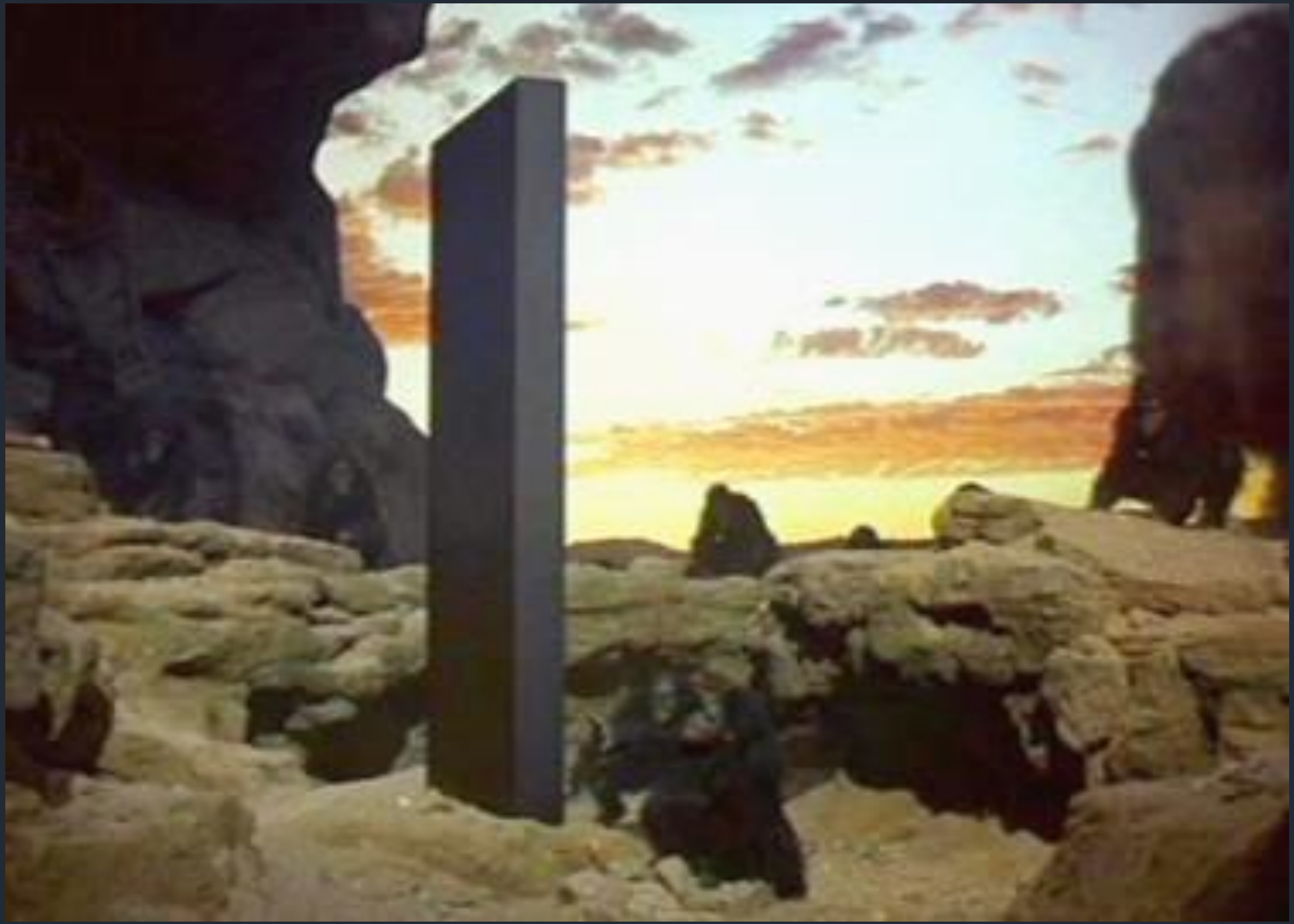
Database transactions have been very fast

because

All changes were LOCAL

So why didn’t all of us heard about this problem already?

DB MONOLITH



By Source, Fair use, https://en.wikipedia.org/w/index.php?curid=31738209

When did we start noticing?

We tried moving this successful DB-based model to “the cloud”.

When did we start noticing?

We tried moving this successful DB-based model to “the cloud”.

Cloud **actually means** ***distributed system.***

Cloud

So my Database is now no longer a Monolith, it is a System. Changes are no longer LOCAL

CAP theorem applies

By Photo by John Kerstholt. original upload by Solitude (From English Wikipedia) [GFDL (http://www.gnu.org/copyleft/fdl.html) or CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0/)], via Wikimedia Commons

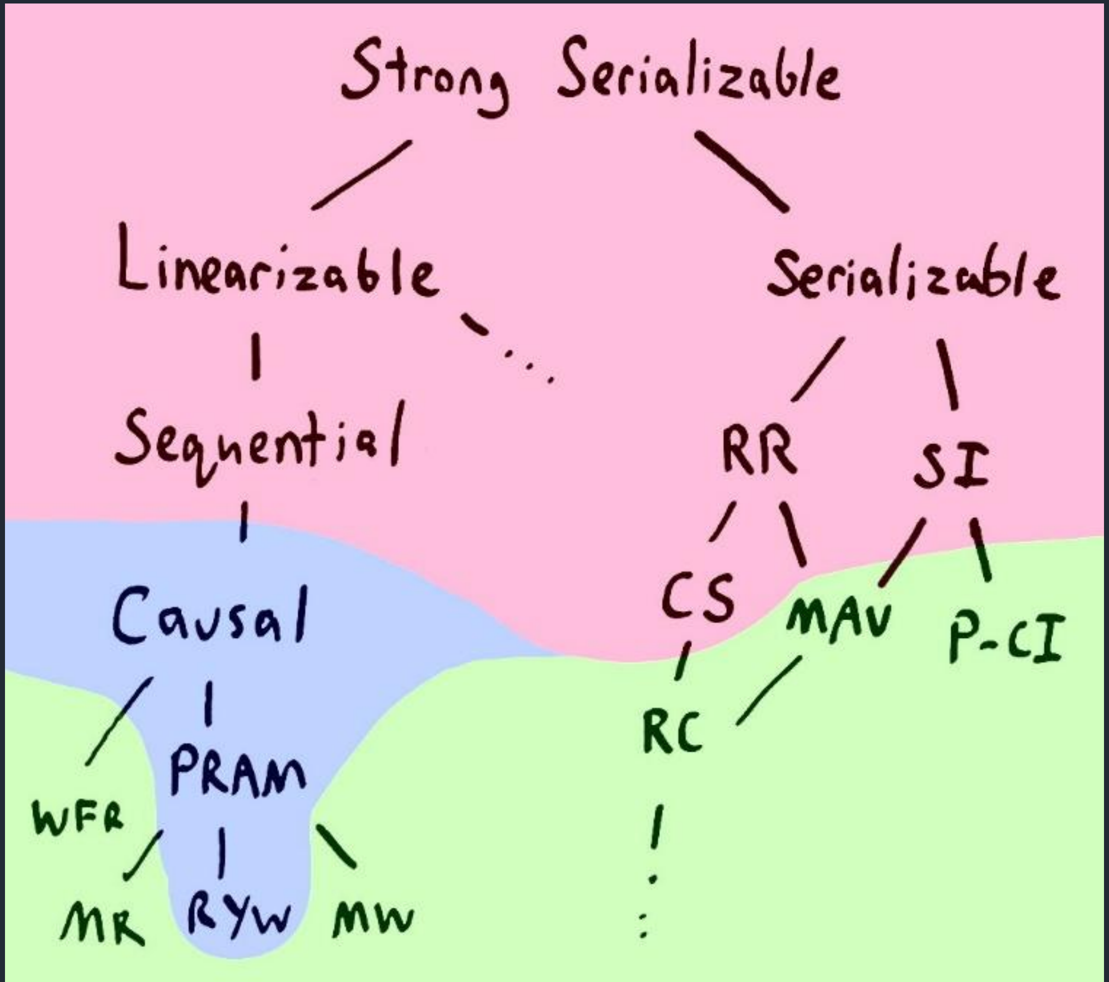
Distributed consistency…

https://aphyr.com/posts/313-strong-consistency-models

http://www.bailis.org/blog/linearizability-versus-serializability/

Linearizability: single-operation, single-object, real-time order

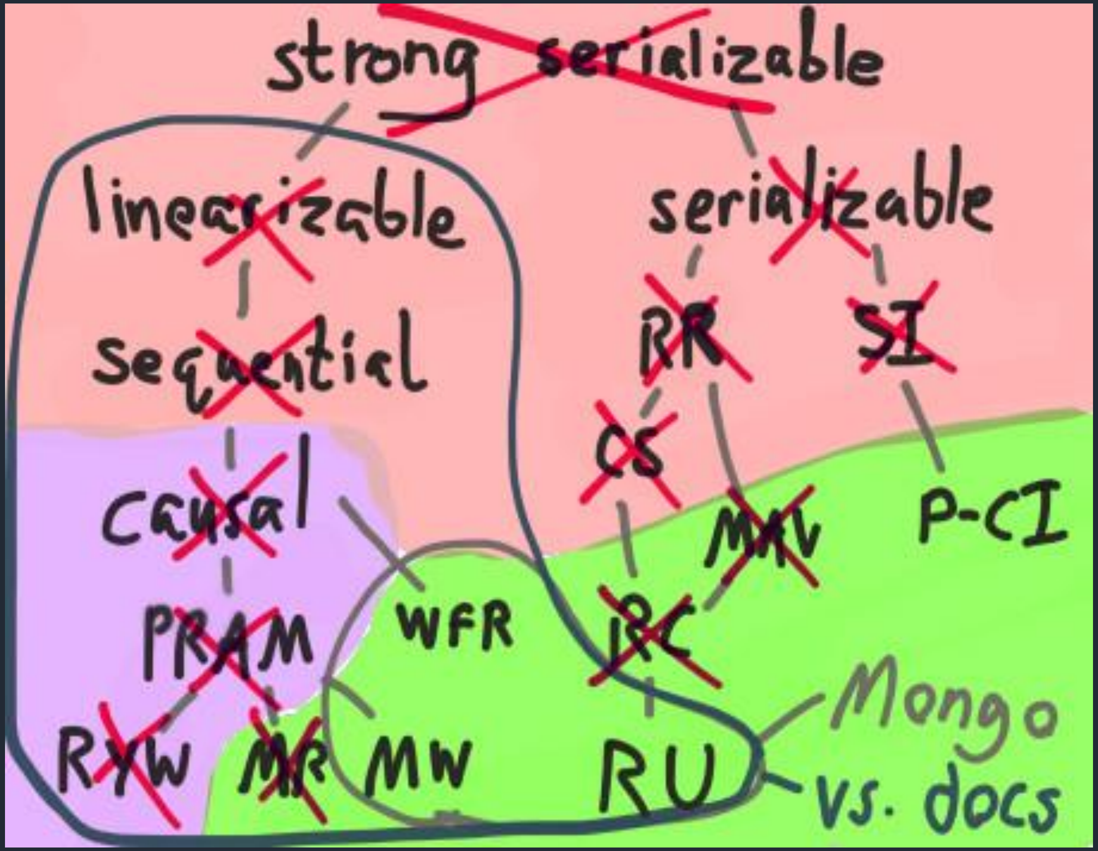
Serializability: multi-operation, multi-object, arbitrary total order



Typical cloud database - MongoDB

https://aphyr.com/posts/322-jepsen-mongodb-stale-reads

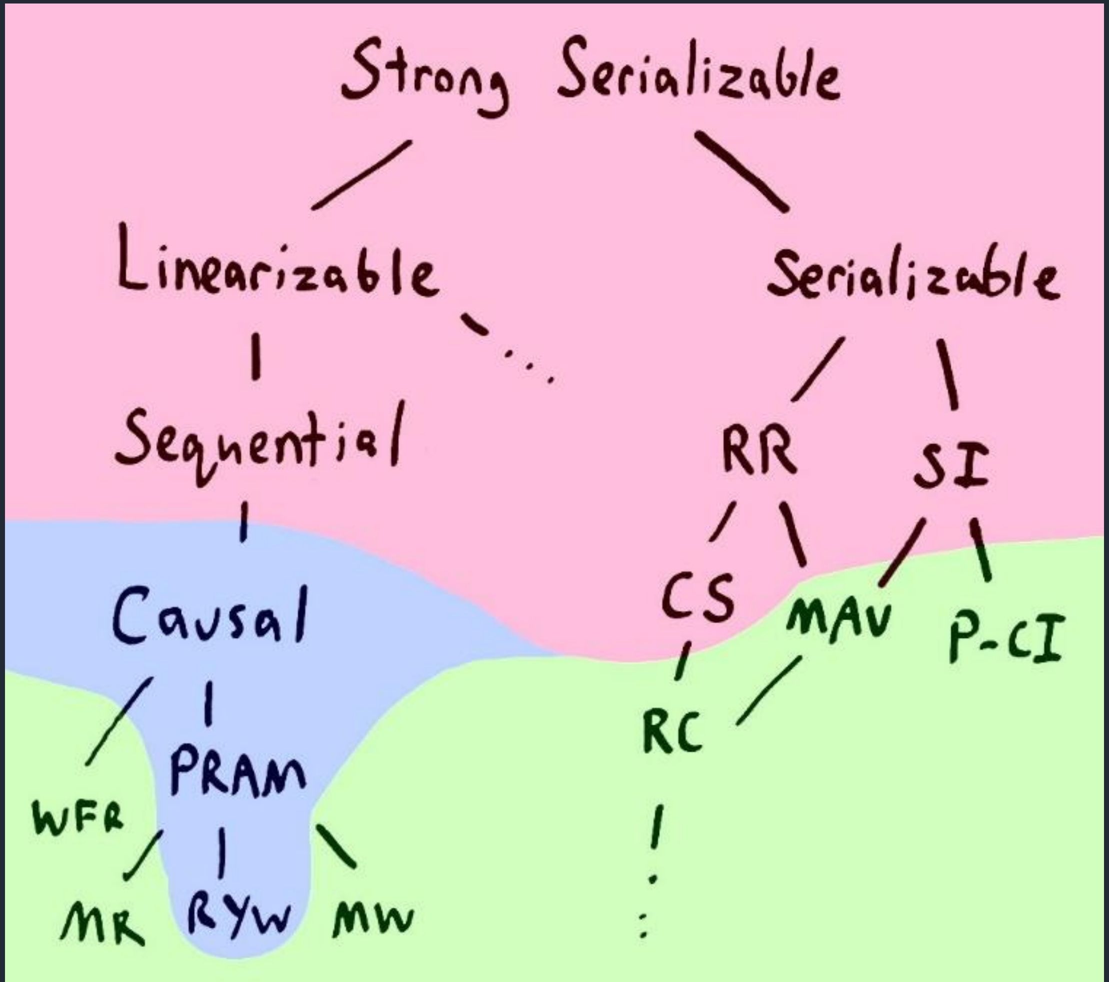
NOT GOOD



Why? – the HAT paper *Highly Available Transactions: Virtues and Limitations – Peter Bailis et al. 2014*

http://www.bailis.org/papers/hat-vldb2014.pdf

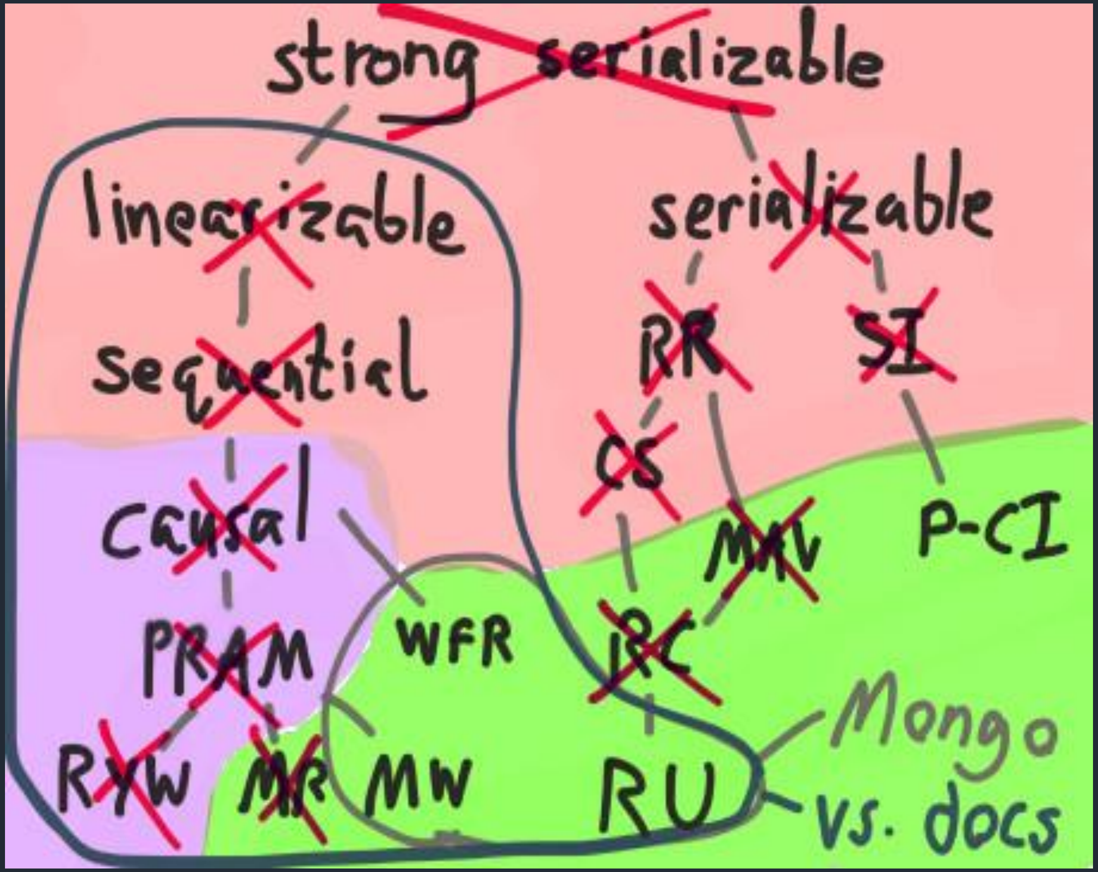
GREEN ones do not impact availability



Why? – the HAT paper *Highly Available Transactions: Virtues and Limitations – Peter Bailis et al. 2014*

When partition occurs, Mongo chooses Availability over Consistency

Besides, being Consistent in Distributed Land is HARD and VERY SLOW

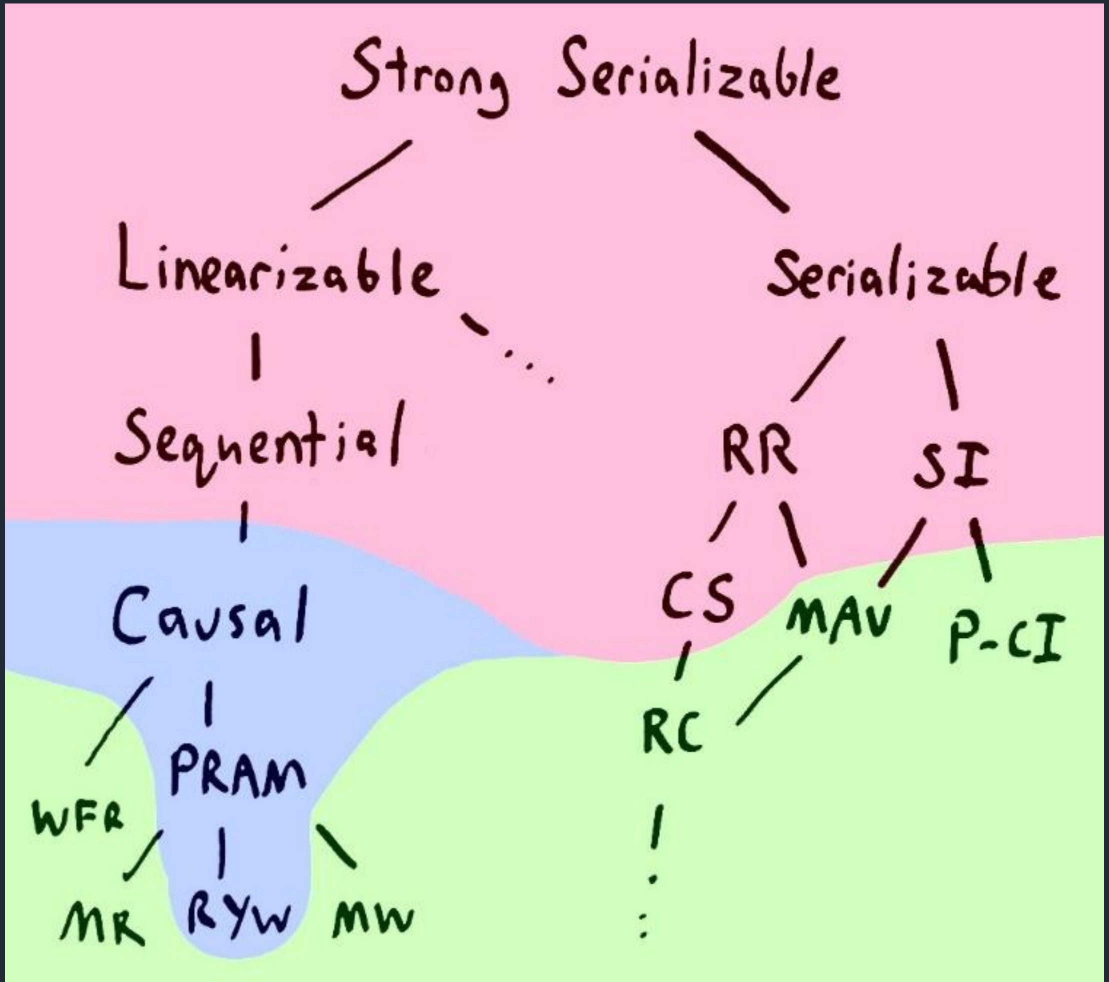


Why?

http://www.bailis.org/papers/hat-vldb2014.pdf

ROSE potential loss of availability

BLUE guaranteed by available system only inside a session

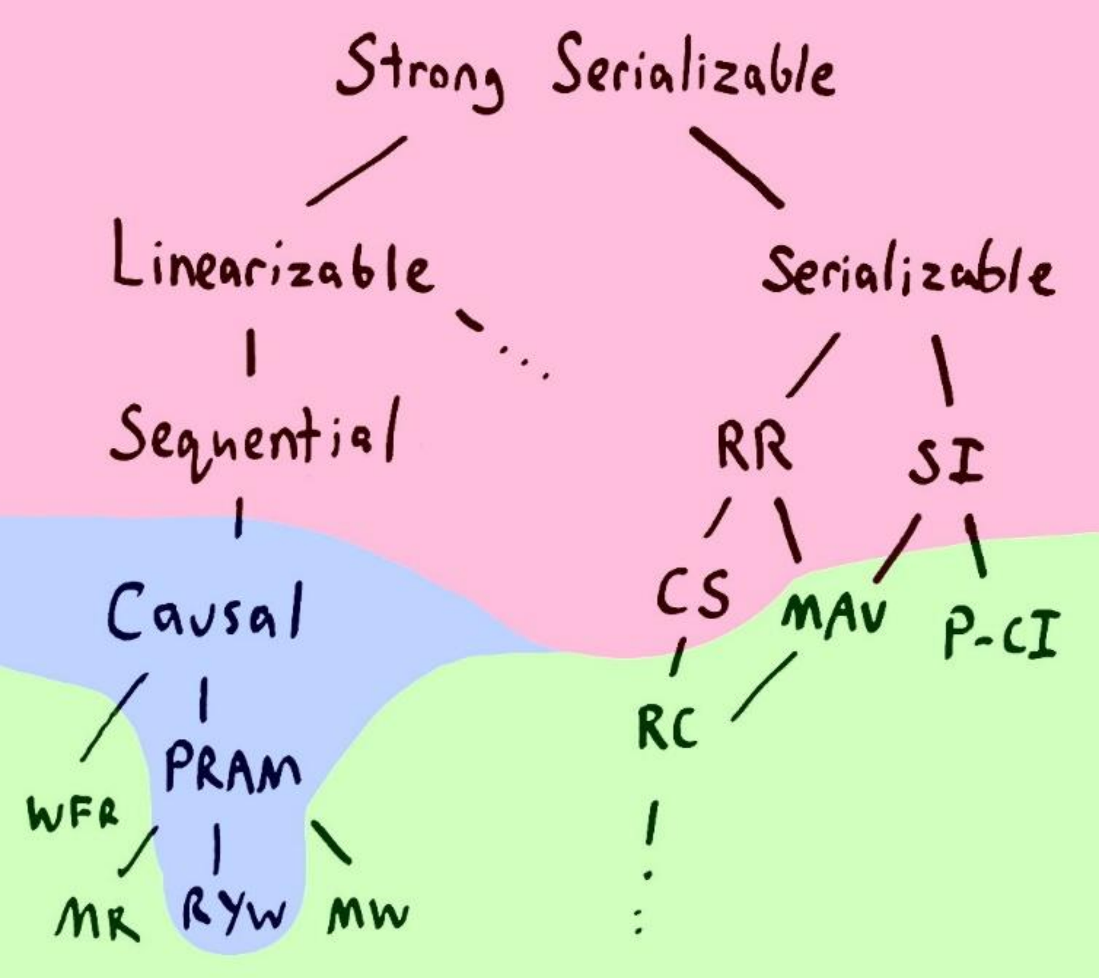


What left and right rose ones have in common?

http://www.bailis.org/papers/hat-vldb2014.pdf

ROSE prohibits Lost Updates

BLUE prohibits Lost Updates inside a session



But Lost Updates were not that bad in Monolith?

Because system was Local everything was fast and Lost Updates were very, very rare...

Plus you can make your system Strongly Consistent and only acceptably slower if you are Local.

Can we get back to Local?

Can we get better?

Can we get back to Local?

Microservices?

Database per Service pattern -
http://microservices.io/patterns/data/database-per-service.html

http://www.lagomframework.com/blog/lagom-1-0.html

lagom

What is the problem now?

Microservice orchestration – how to control the emergent behaviour of group of microservices performing a chain of transactions.

When partition heals, we either merge results or take back affected operations.

Merging ☺

https://chuva-inc.com/blog/fast-tip-enable-git-rerere-right-now

https://git-scm.com/docs/git-rerere

```
the number of planets are
<<<<<<< HEAD
nine
=======
eight
>>>>>>> branch-a
```

I DON'T ALWAYS HAVE MERGE CONFLICTS
BUT WHEN I DO I HAVE NO IDEA HOW TO SOLVE

Doing merges is hard.

And requires domain knowledge.

Sagas – *Hector Garcia-Molina, 1987*

Also: http://kellabyte.com/2012/05/30/clarifying-the-saga-pattern/ and "ACTA: The SAGA Continues" – Chrysanthis and Ramamritham.

Saga - one long running transaction consisting of many transactional steps - pair each transactional operation with its compensation.

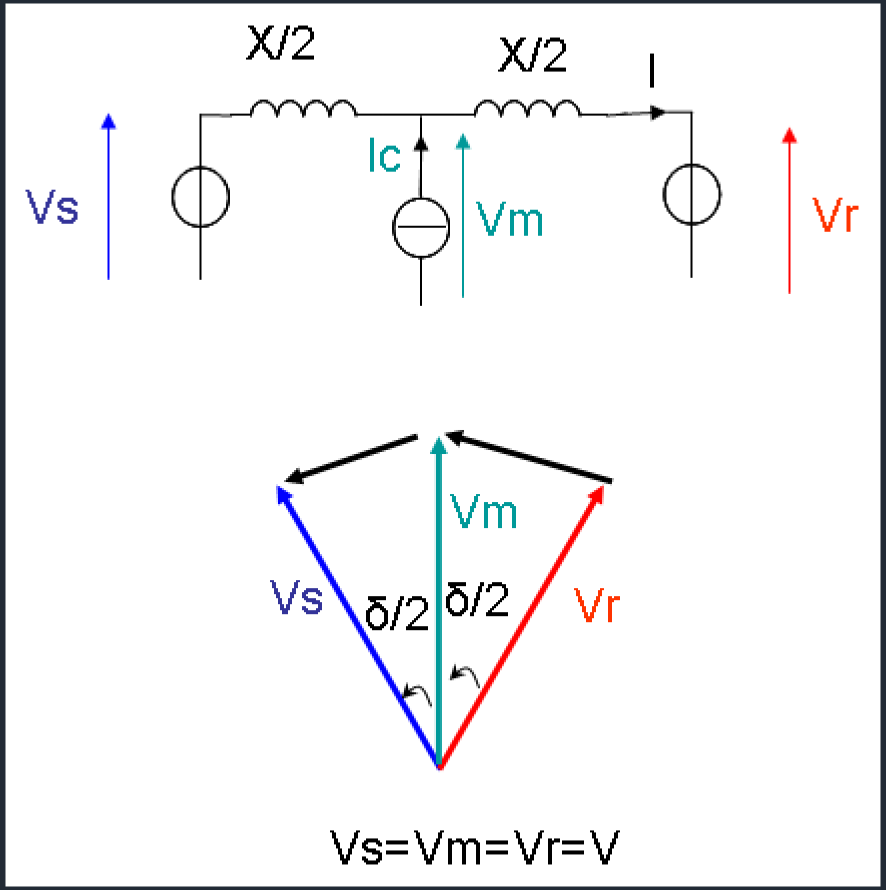
L: T1 → T2 → T3 → T4

C1 ← C2 ← C3 ← C4

Sagas – *Hector Garcia-Molina, 1987*

Also: http://kellabyte.com/2012/05/30/clarifying-the-saga-pattern/ and "ACTA: The SAGA Continues" – Chrysanthis and Ramamritham.

We just need to define what should the compensation steps comprise of.



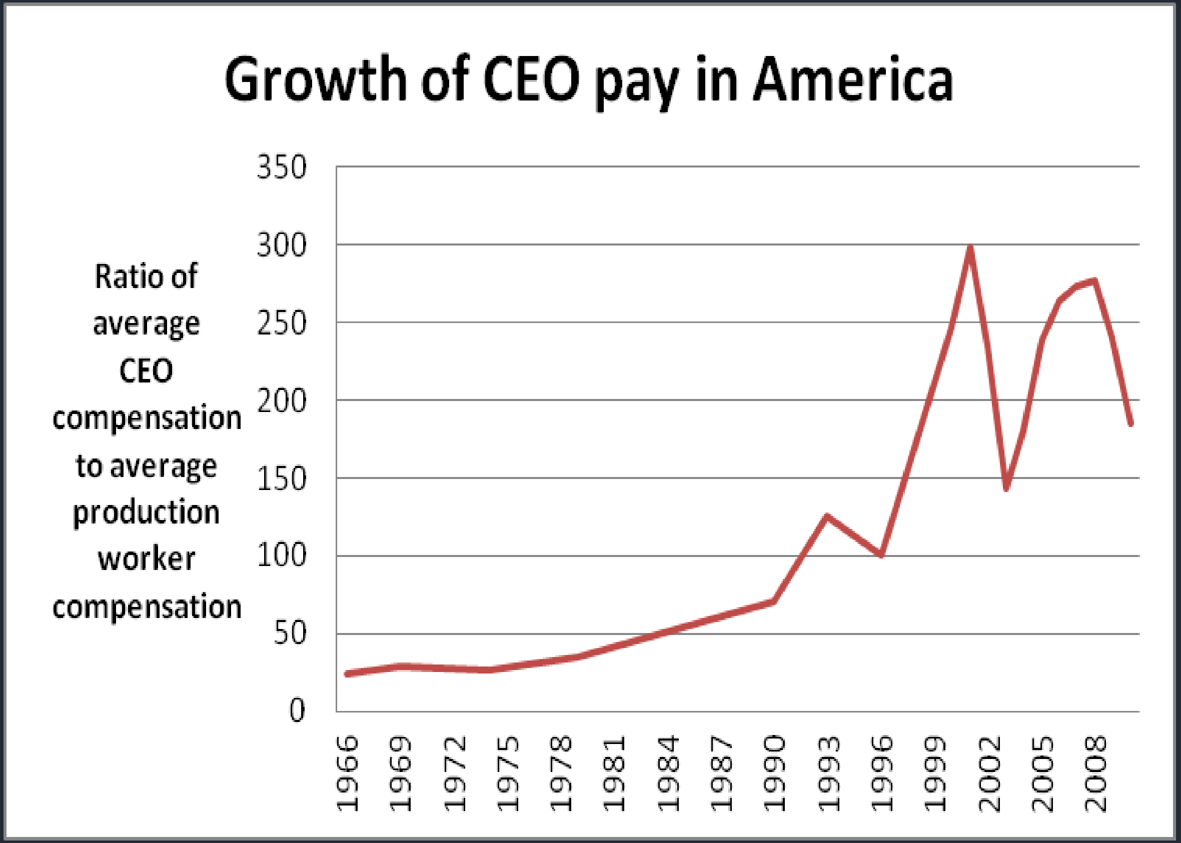
Shunts compensation - By Hermione12 - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=2509621

Sagas – *Hector Garcia-Molina, 1987*

Also: http://kellabyte.com/2012/05/30/clarifying-the-saga-pattern/ and "ACTA: The SAGA Continues" – Chrysanthis and Ramamritham.

What is a good compensation?

Is the compensation possible at all?



Exercise: What is the right level and manner of compensation?

Cases:

- **Missing transactions in the Web Banking app**
- **Knightmare on Wall Street**
  - https://dougseven.com/2014/04/17/knightmare-a-devops-cautionary-tale/
- **Multiplicated Visa Charges (up to 15x)**
  - http://prawo.vagla.pl/node/7663

Exercise: What is the right level of compensation?

Cases:

- **Missing transactions in the Web Banking app**
- **Knightmare on Wall Street**
  - https://dougseven.com/2014/04/17/knightmare-a-devops-cautionary-tale/
- **Multiplicated Visa Charges (up to 15x)**
  - http://prawo.vagla.pl/node/7663

Final words

When we allow inconsistency it is always a Business level decision what to do.

So it is a Functional Requirement

We need to communicate this clearly.

Thank you!

Questions?