

# An Erlang Game Stack



**Erlang User Conference 2012**  
**H. Diedrich**

# Henning Diedrich

- Maintainer Emysql, Erlvolt, Erlualib
- Eonblast: *game + film*
- NewTracks: *game + music*
- Bricks and mortar
- Insurance tariff language

Octocat says: try markedoc!





# EONBLAST

WELCOME TO THE 30th CENTURY

HOME  
GAME  
WORLD  
STORY  
ART  
TECHNOLOGY  
OPEN SOURCE  
MISSION  
COMPANY

**Get the book!**



## WELCOME TO THE 30TH CENTURY

Eonblast is a new studio that **pushes the envelope** of what can be achieved in games.

Imagine the **next level** of browser based Massively Multiplayer Online Games (MMOG), both **visually**, **narratively**, and **technologically**. Then put **roleplay (RP) back** into the game and you're close.

Imagine the power of the **Cloud** crossed with emerging new generation of **Alternate Reality** games: players empowered to actually\* **change their world**.

We are combining the crafts of **games** and **film** to create a new, **more immersive experience**.

\*as opposed to make believe.



# This Talk

- A game server architecture
- Its components
- The glue

# This Talk

- Work in progress
  - Unproven concepts
  - Lies and conjecture
- andalso**
- 3 genius tools
  - 10,000 lines of glue code

# Questions

Please throw them in like Torben  
Mail me at [hdiedrich at eonblast.com](mailto:hdiedrich@eonblast.com)  
Erlang Mailing List

# Erlang Game Servers



Zynga: FarmVille via membase, Activision Blizzard: Call of Duty, Bigpoint: Battle Star Galactica, Wooga: Magic Land

# Erlang + MySQL



**RUPTURE**



<http://eonblast.github.com/Emysql>

Electronic Arts contributed emysql.



# The Hurt

- Leaks, Deadlocks
- Bad Productivity
- Game Design Language Gap
  
- Less Features
- Less Players
- Lesser Product

# Target

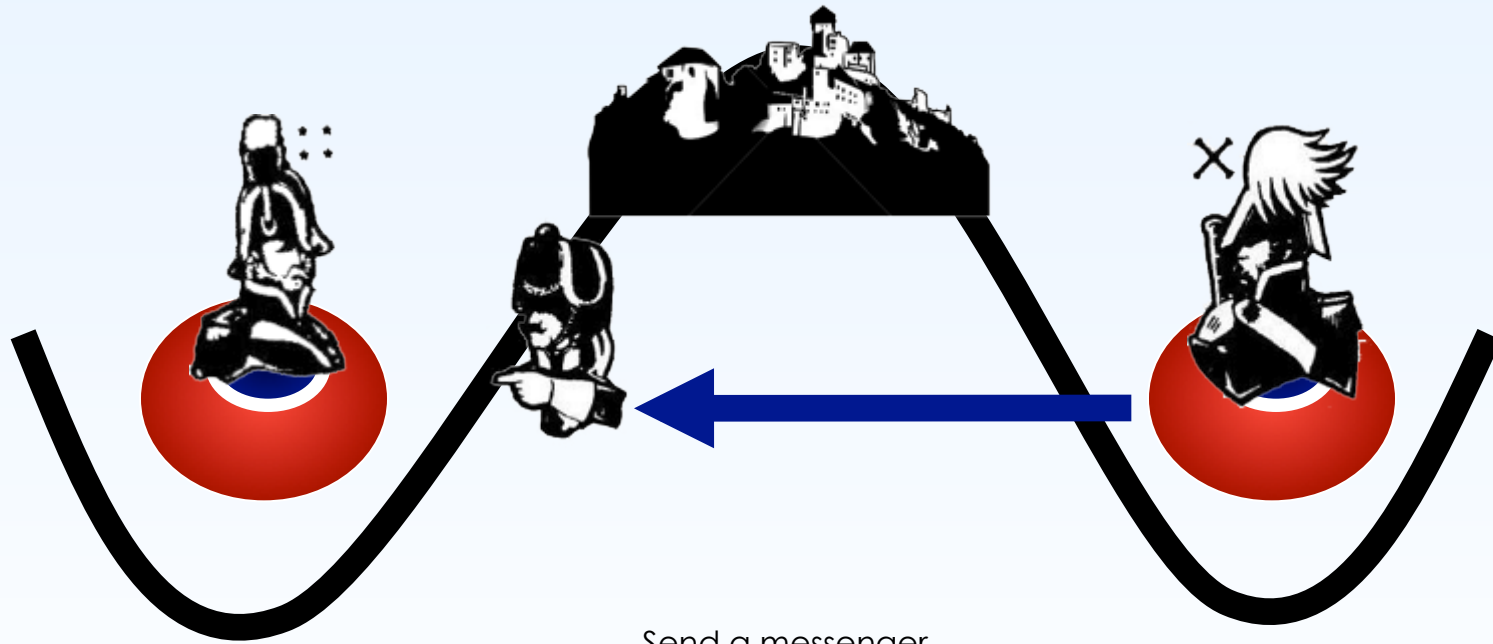
„The perfect game server“  
(makes no sense)

# Generals' Problem



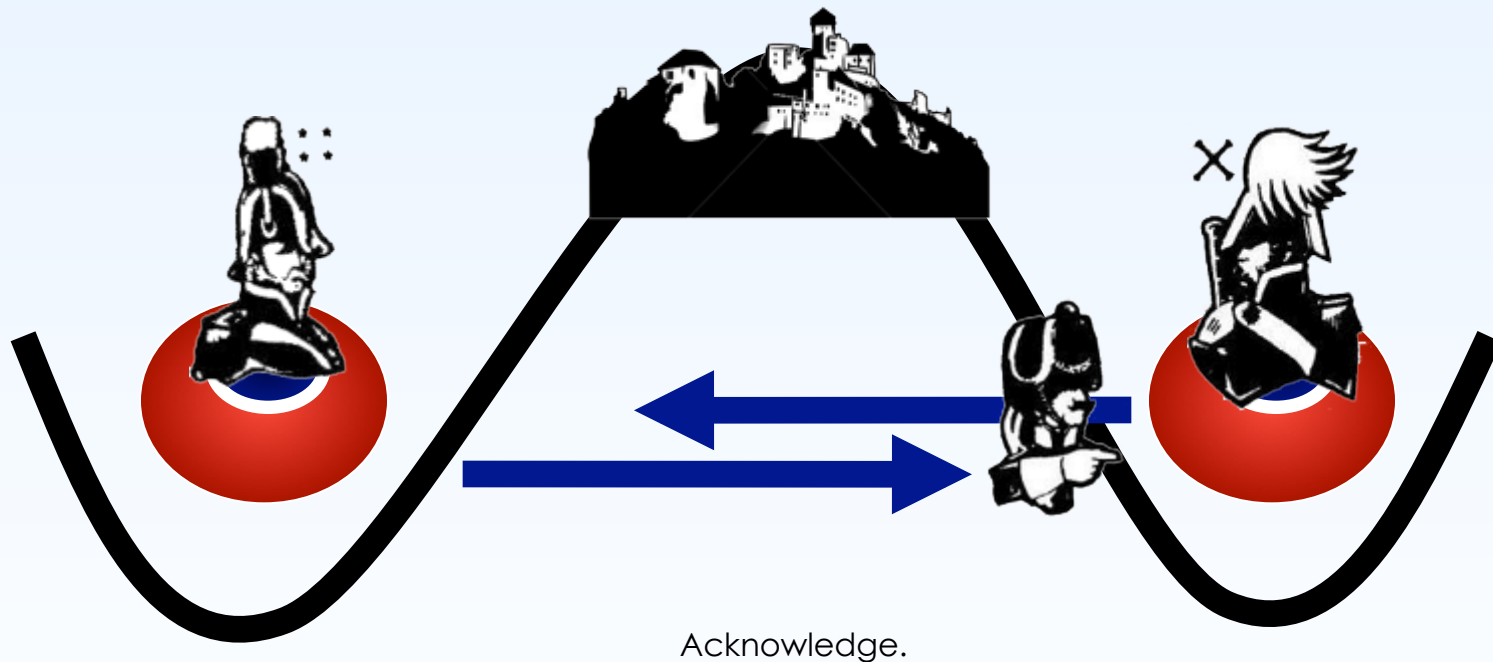
Two generals must agree on a time.

# Generals' Problem

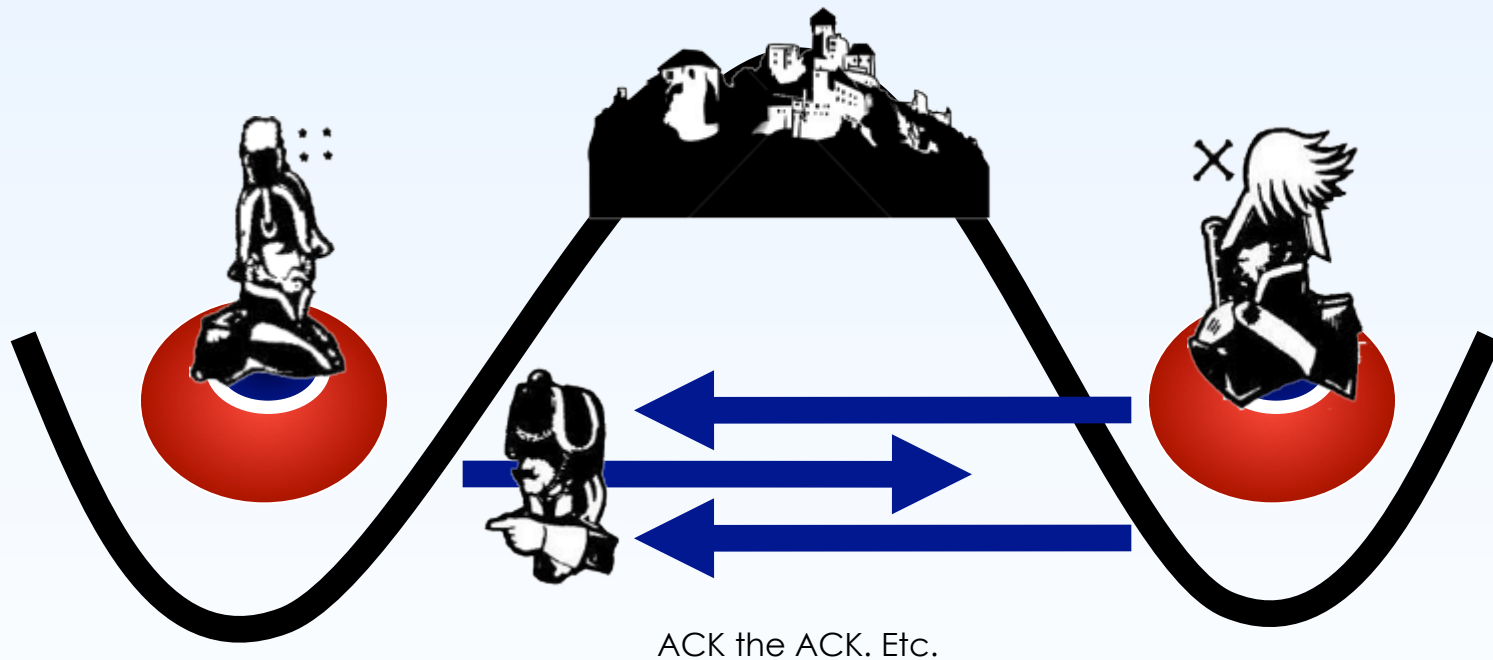


Send a messenger.

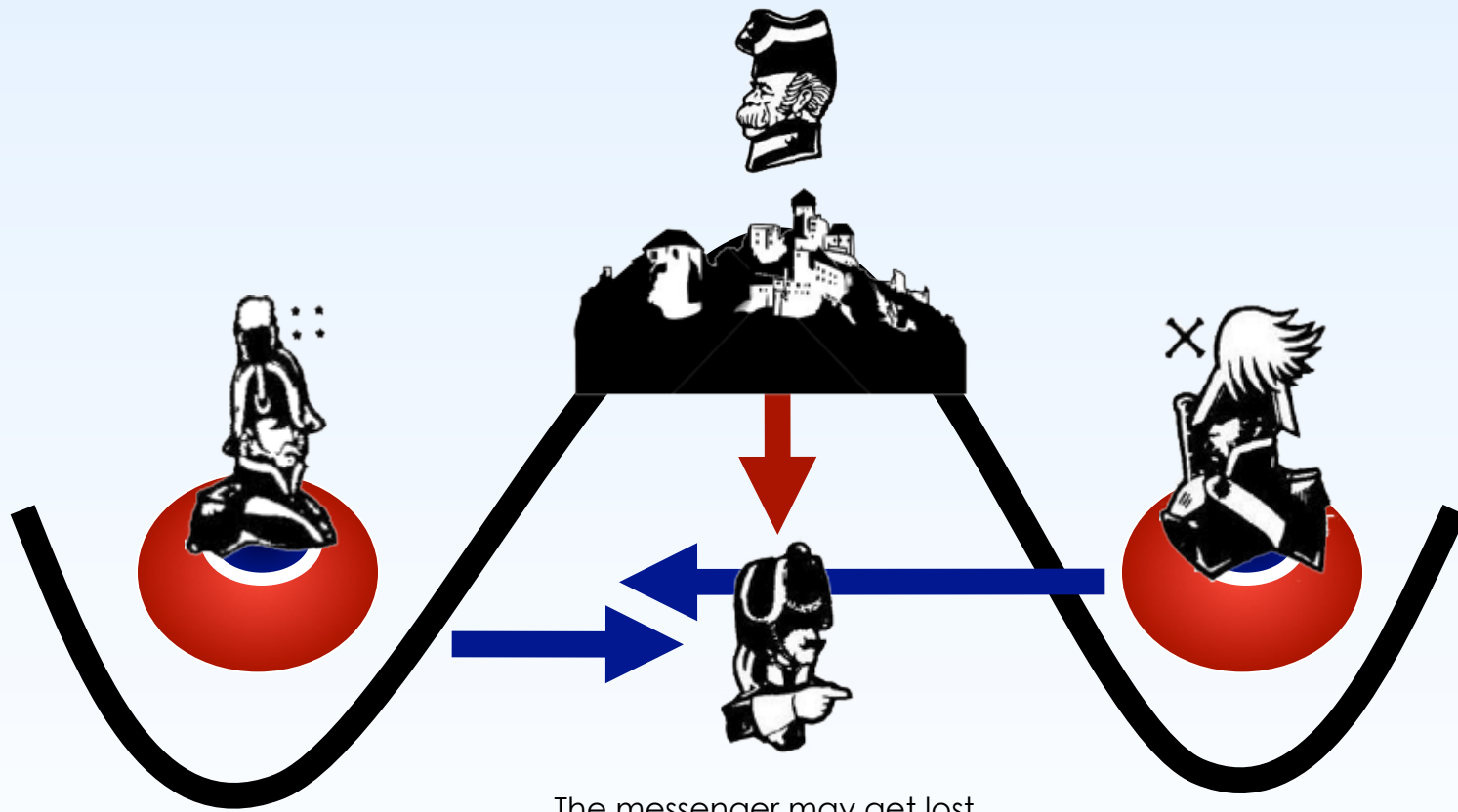
# Generals' Problem



# Generals' Problem

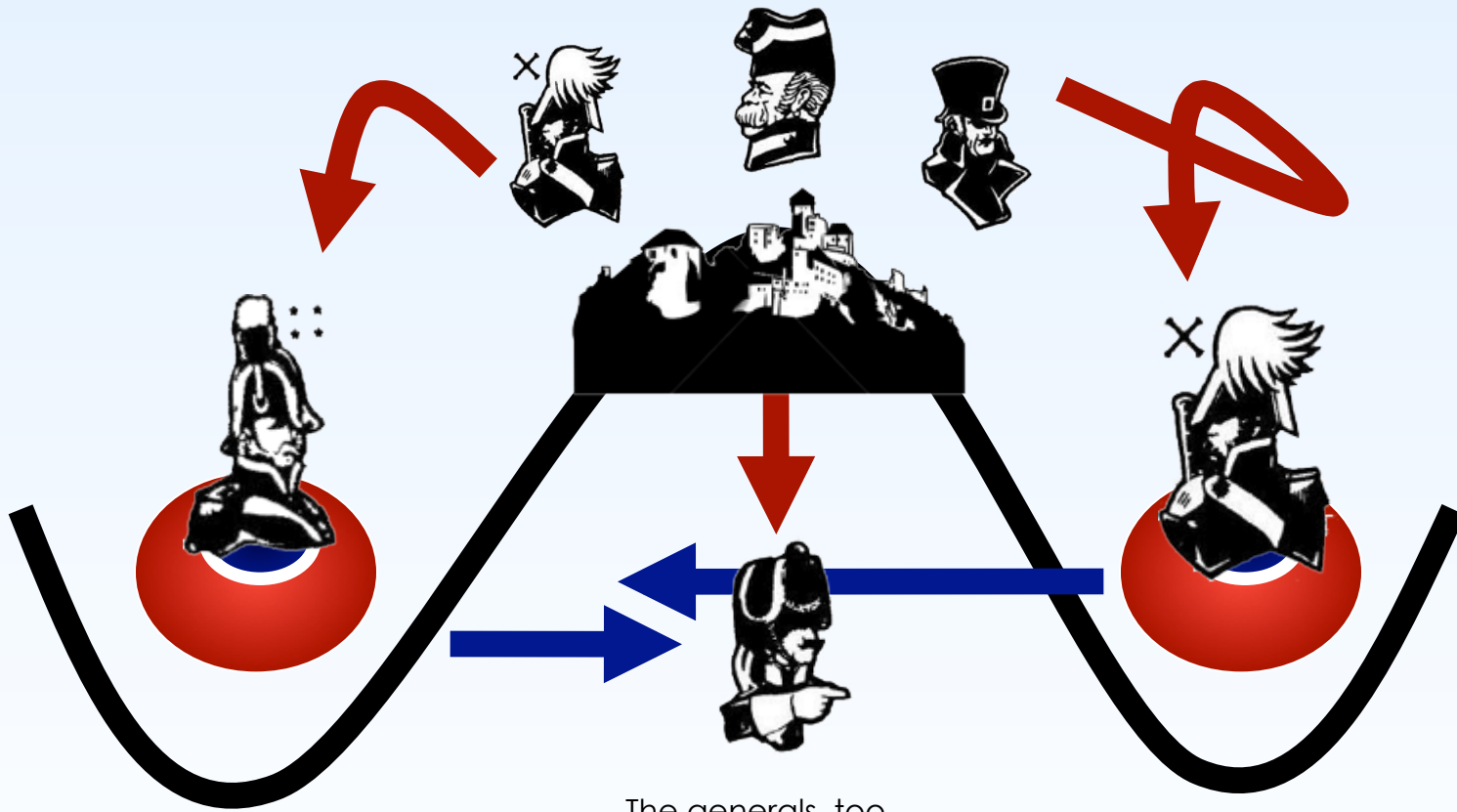


# Generals' Problem



The messenger may get lost.

# Byzantine Generals



The generals, too.



# The Hunt

## **Server Stack**

- Language
- Database
- Protocol
- Game Logic

# Spec

## “Table Game”

- 1 million active players
- 1 million transactions per second
- 1 second latency
- Linear scale
- 100% data integrity

# Client Architecture

- Fat JavaScript Browser Client
- COMET

Not discussed in this talk.

# Language

## Wishlist

- Robustness
- Deadlock-free
- Multi-core
- Linear Scale

er?

# Database

## **CAP**

- Distributed
- Consistent
- Highly-available
- Partition-tolerant

**All of it!**

Brewer on CAP 2012: <http://www.infoq.com/articles/cap-twelve-years-later-how-the-rules-have-changed>

# Database

## **ACID**

- Atomicity
- Consistency
- Isolation
- Durability

**For Granted?**

# Database

## **Double Bookkeeping**

- Not every game needs it
- Requires ACID Transactions
- Neigh impossible to emulate
- Impossible with BASE /  
Eventual Consistency

# Database

## Upcoming Blog Post

- Looking at 14 databases
  - Riak, Cassandra, Membase, H-Base, Voldemort, MySQL, MySQL Cluster, Redis, Redis Cluster, Tokyo Cabinet, Memcached, CouchDB, Couchbase, VoltDB, MongoDB, Postgres
- In the light of what games need
- Unbiased comparison
- Twitter @hdiedrich



# Protocol

## **Wishlist**

- Small footprint
- Fast to encode
- Human-readable

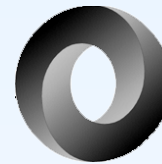
# Logic

## **Wishlist**

- Human Readable
- Stable
- Fast
- Concise
- Small Footprint

# The Answers

- Client
- Protocol
- Server
- Logic
- Data



Client: HTML5/JS, Protocol: JSON, Server: Erlang, Logic: Lua, Data: VoltDB.



**“Embeddable scripting language”**



# Lua

*"Lua is strange."*  
R. Viriding

"It has mutable variables, what more need I say."



# Lua

```
a = "Roberto"  
if a ~= nil then  
    print "Hello " .. a .. "!"  
end
```



# Lua

- Created 1993
- Simple, imperative syntax
- Stable syntax, stable VMs
- Two main VMs: ISO C, JIT
- Game scripting champion
- Open Source, 100% dictatorial\*

\*benevolent, of course

Site: <http://www.lua.org>

List: [lua-l@lists.lua.org](mailto:lua-l@lists.lua.org)



# Lua

- Made for non-programmers - imperative syntax
- Powerful for programmers - Scheme semantics
- Made for DSLs - extensible semantics
- Made to be extended - simple C API
- Made to be embedded - small VM footprint
  
- Now with a native Erlang VM!





# Lua Caveats

- Deceiving Looks
- Encourages magic
- Simple but not for beginners
- Syntax forked
- Only one collection type



# Lua VMs

- **PUC Reference VM**

Lua 5.2

pure ISO C  
super compatible

---

- **JIT VM**

remains at Lua 5.1  
adds extensions

X86, ARM, PPC, MIPS  
super fast

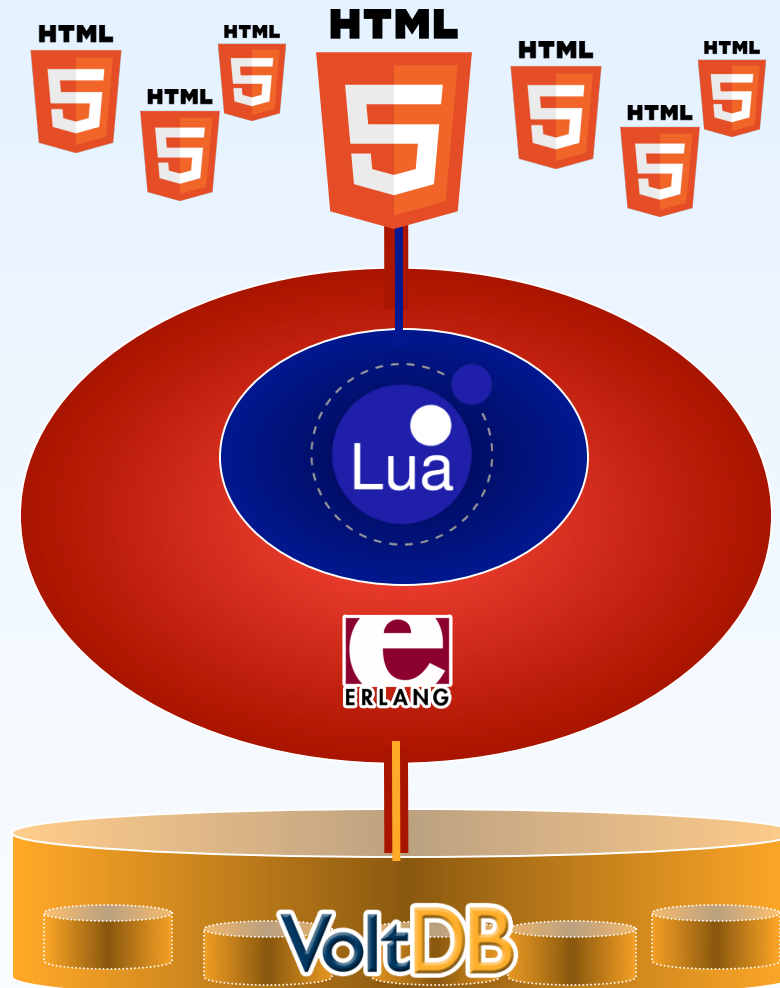
---

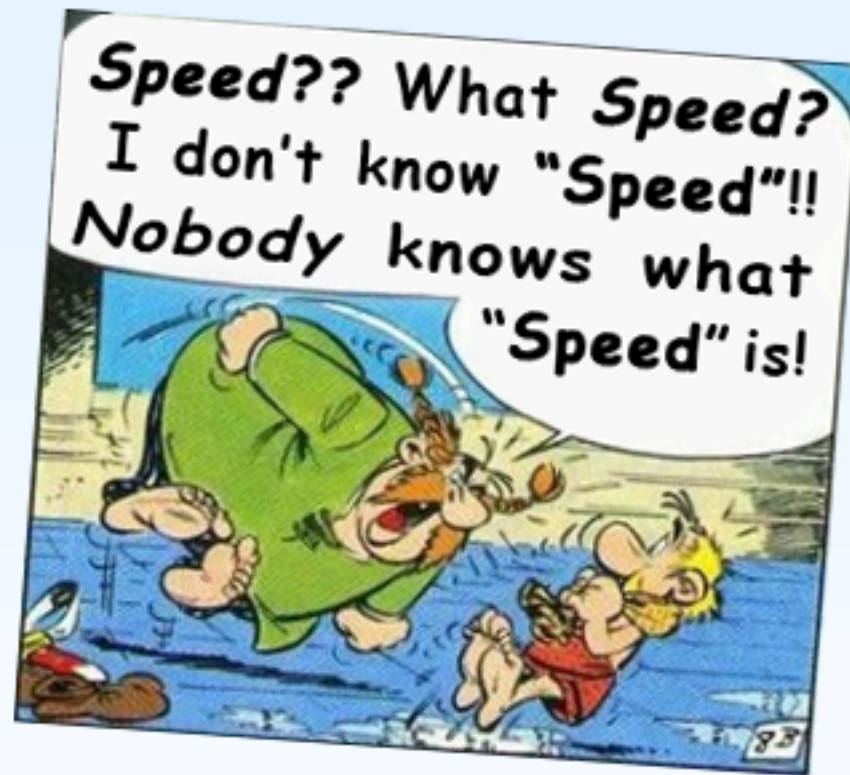
- **Luerl** 

Lua 5.2  
incomplete

native Erlang VM  
super stable

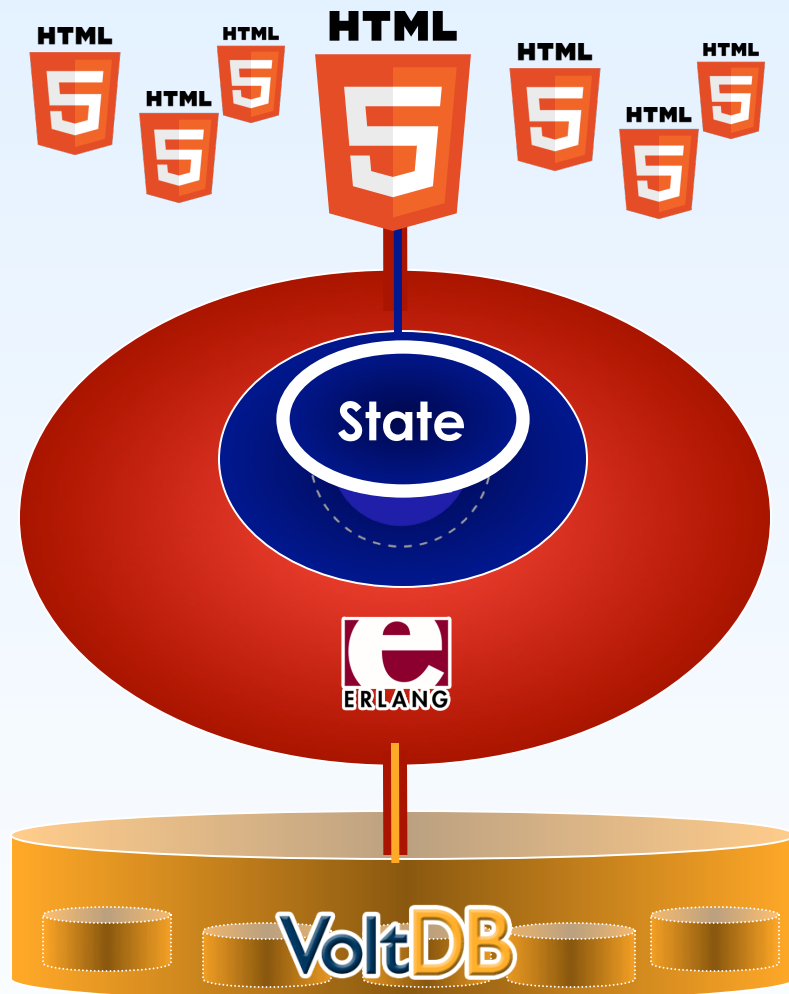
# Architecture





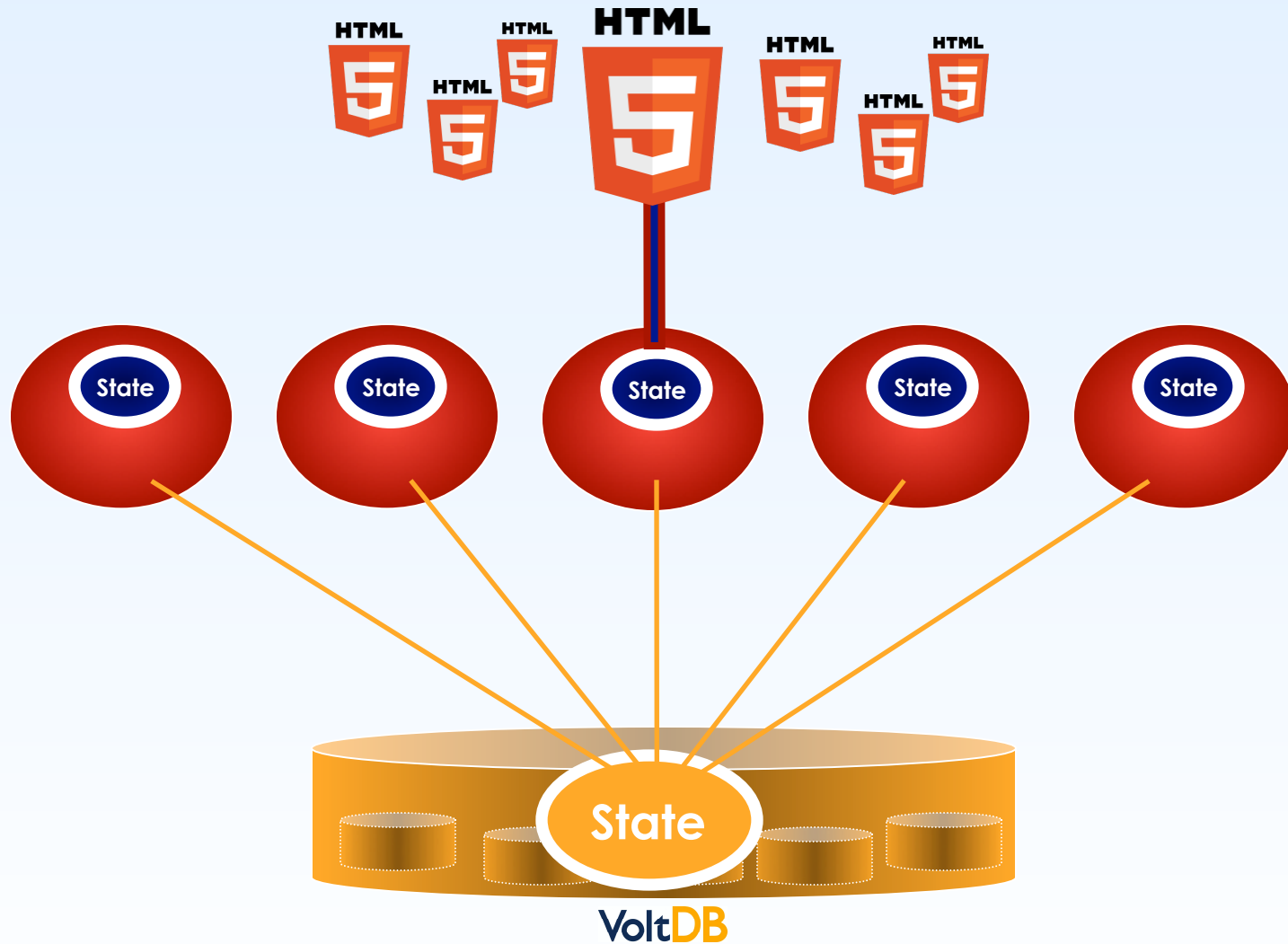
Perceived reaction to asking about benchmarks on the Erlang mailing list.

# World State

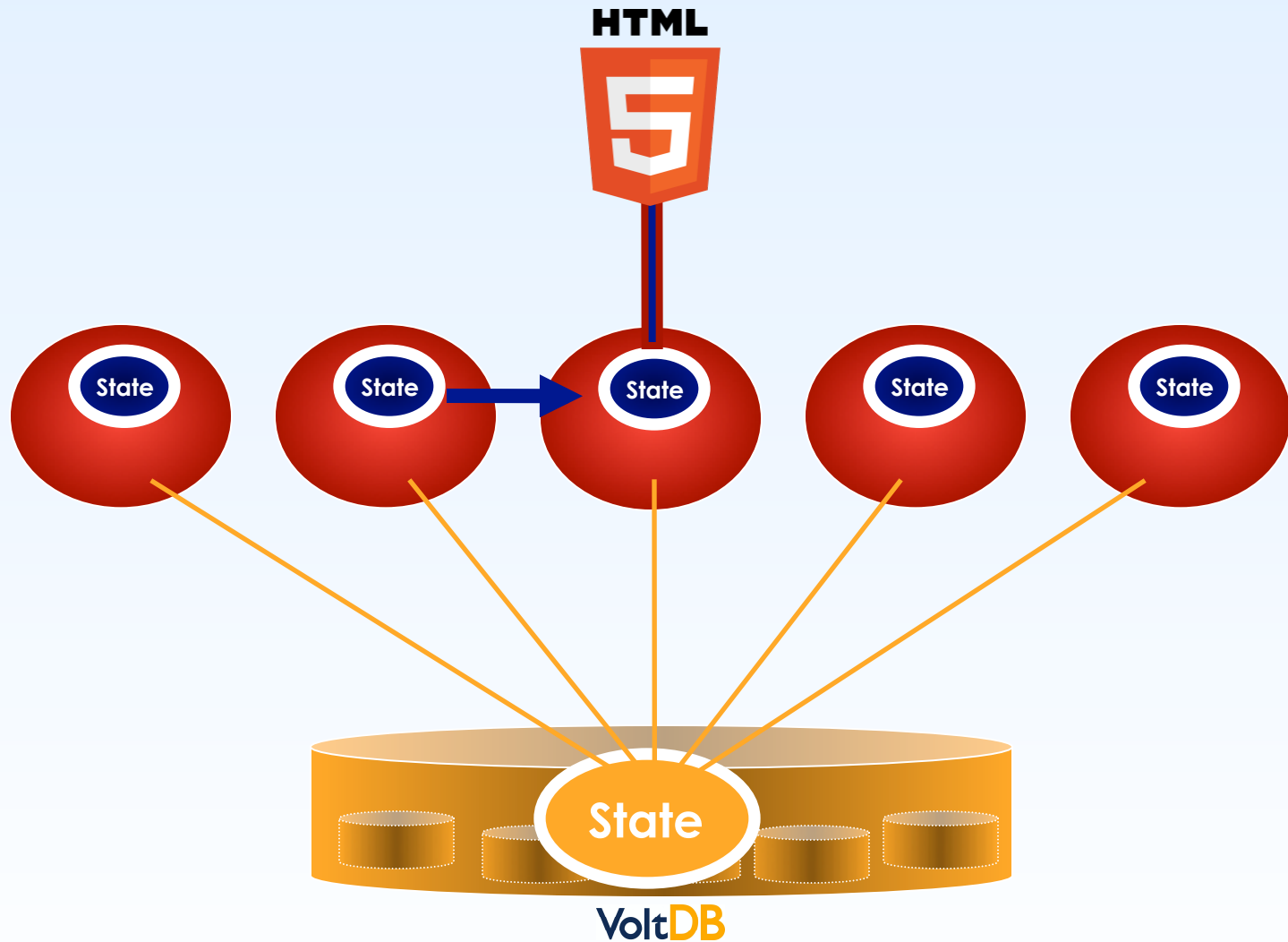


Where does the state go? Into the Logic core, for speed.

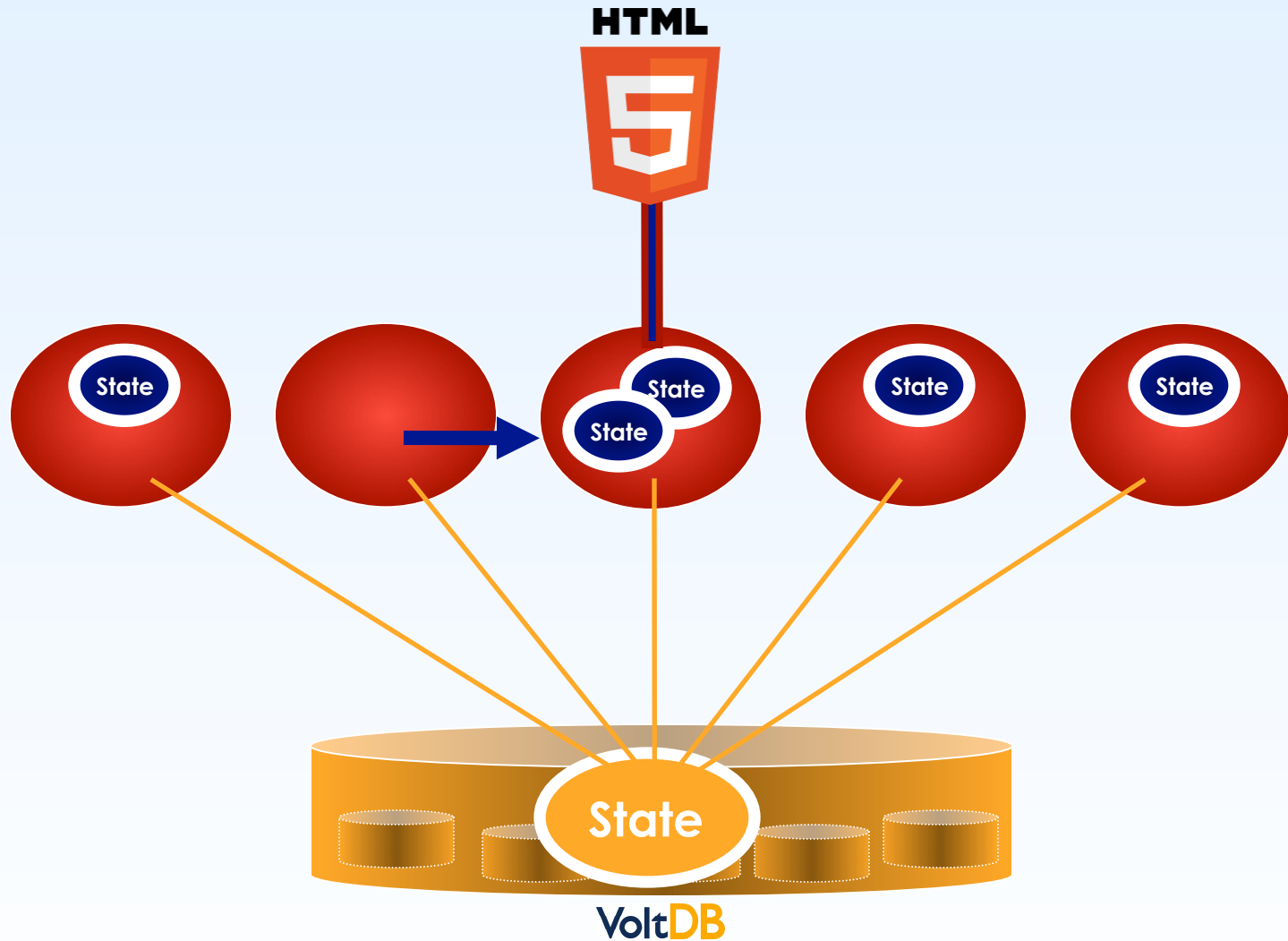
# World State



# World State

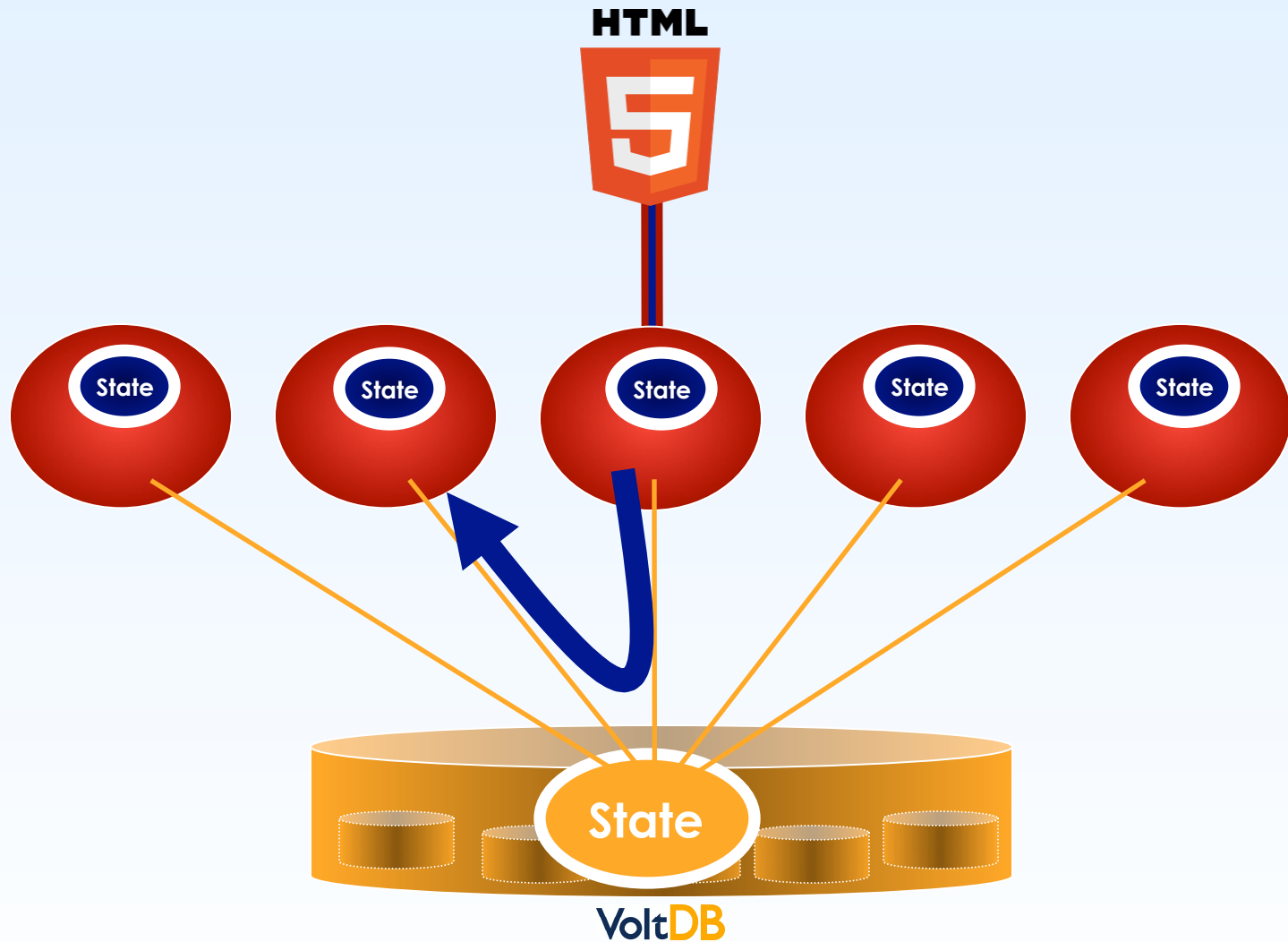


# World State





# World State



**VoltDB**

**“High Velocity Database”**



# VoltDB

- Created 2009
- Simple SQL syntax
- Stable 2.5
- Commercial developer, support
- High Velocity Secret Tip
- Open Source, 100% dictatorial\*

\*benevolent, of course

Site: <http://www.voltdb.com>

Help: <http://community.voltdb.com/forum>



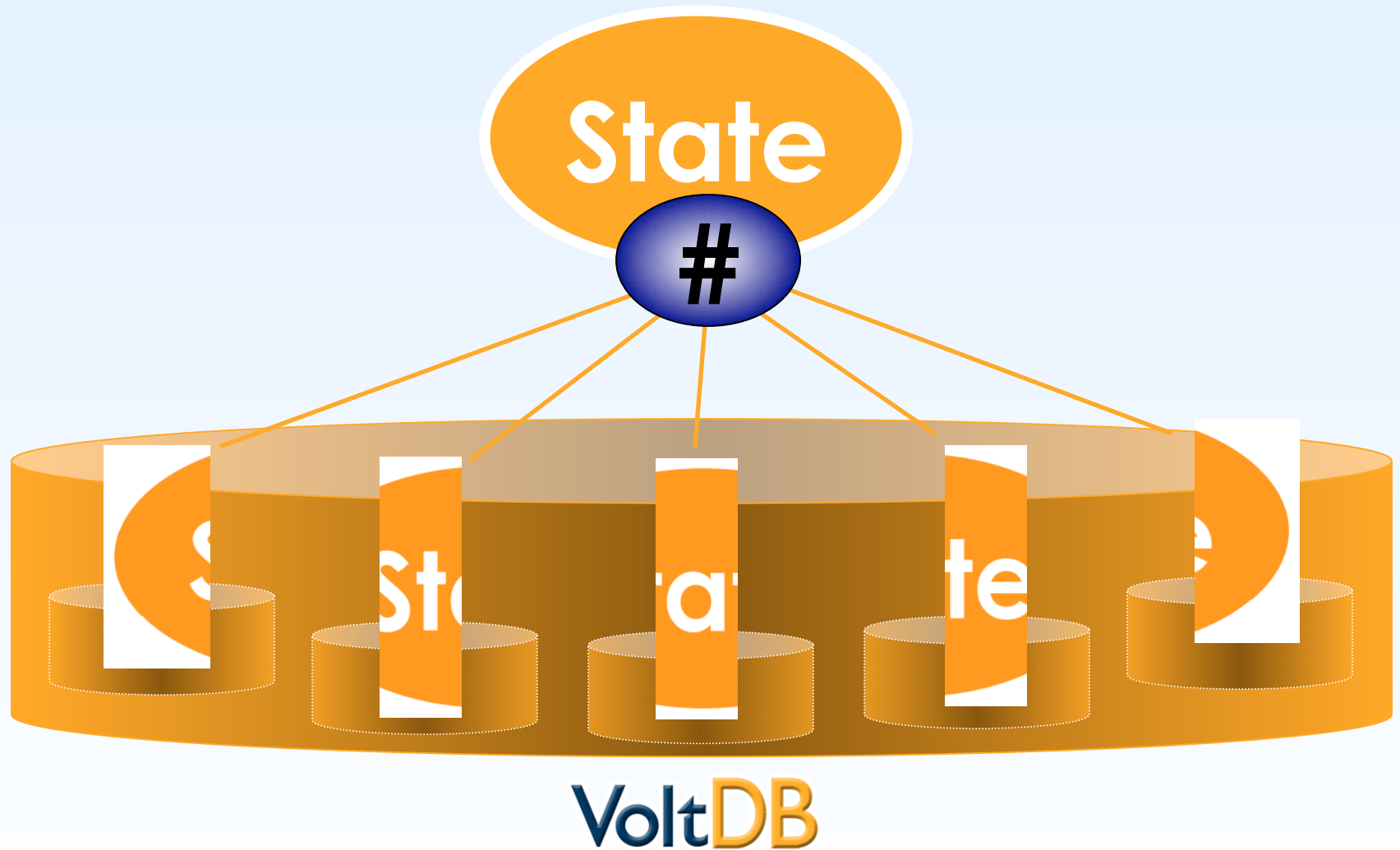
# VoltDB

- ACID transactions – double bookkeeping
- SQL – subset of '92
- Linear scale - „unlimited“ data (but ...)
- Made for OLTP - fast, cheap writes, high throughput
- “More SQL than SQL” – invites clean sep. of data
- In-memory - 100x faster than MySQL
- Replication, Snapshots – 'hot backup built in'

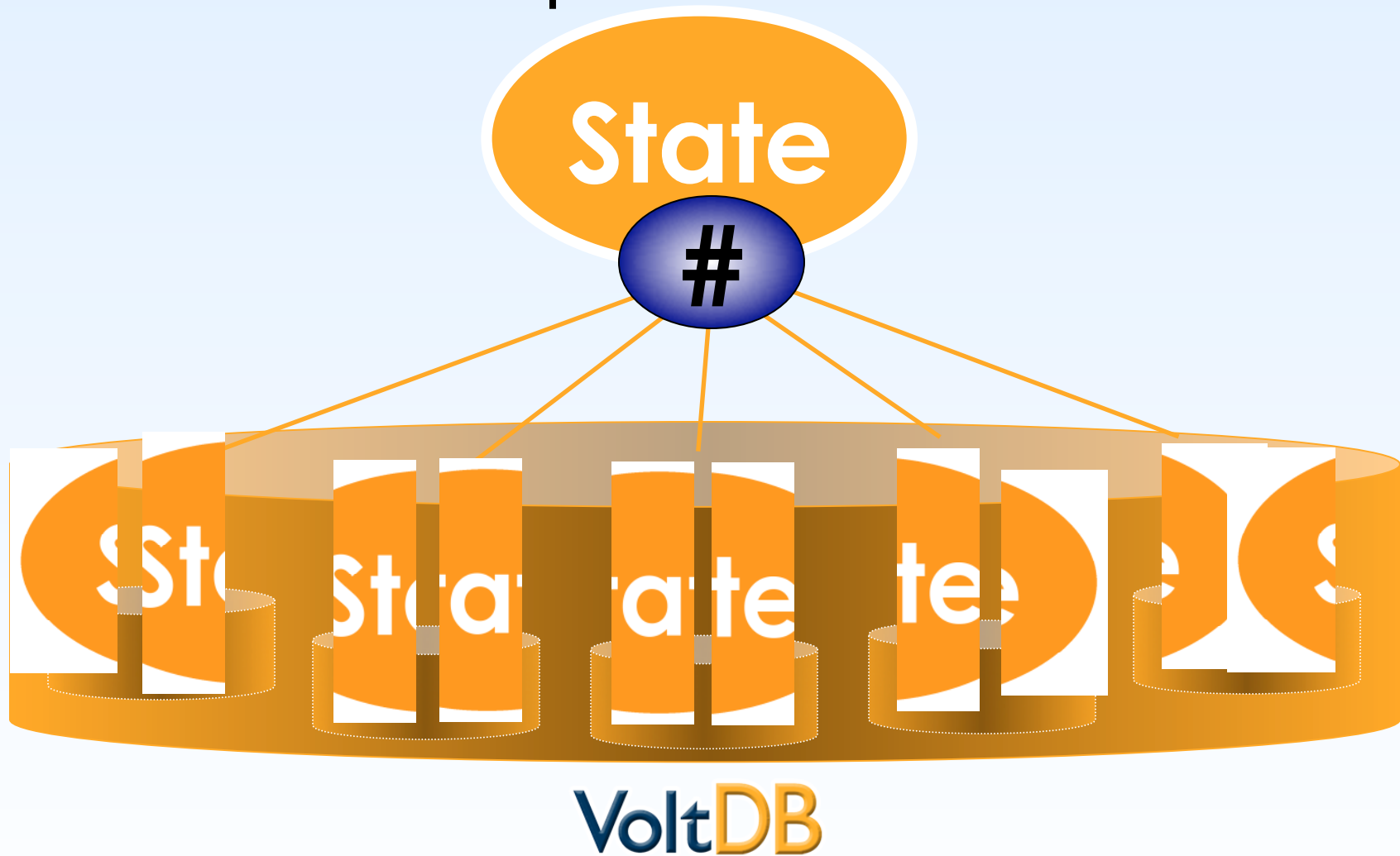
# VoltDB VoltDB Caveats

- Less agile than MySQL/NoSQL
- Need to understand partitions
- Must program SPs in Java
- Not elastic (yet)
- No online schema change (yet)
- There is always a bottleneck
- Simple but not for beginners

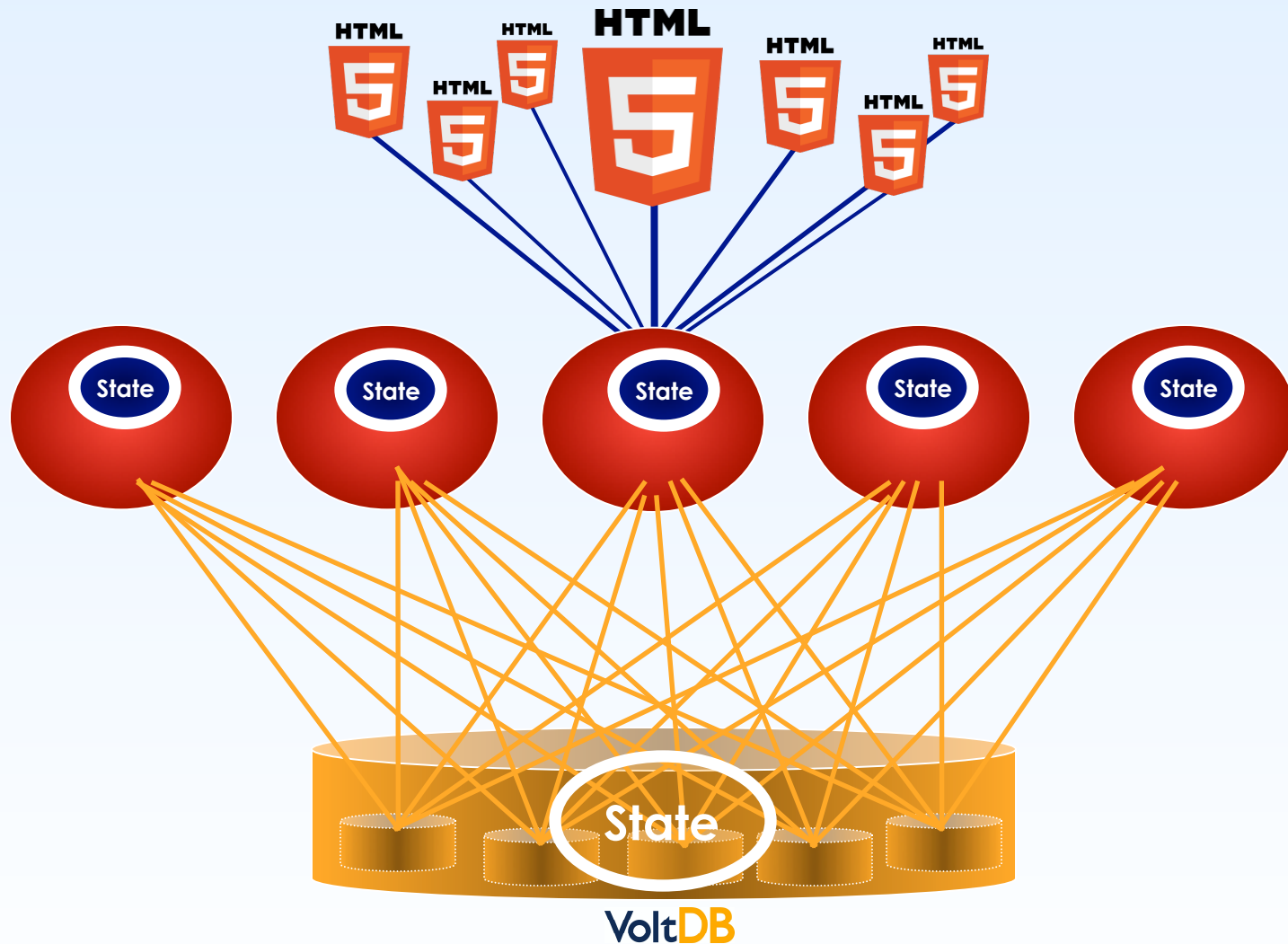
# Horizontal Partitions



# Replication



# Actual Connections





# The Mix















Erlang + Lua + VoltDB







# The Packages

- Protocol  Lua to JSON  
- Logic  Lua port  
- Logic  Lua VM  
- Data  VoltDB driver  



# Erlualib



- Lua *embedding* library for Erlang
- Lua state in original PUC Lua VM
- A fast way
- A not secure way: can crash Erlang



# Erlualib



```
hello() ->
  % get handle of the Lua engine
  {ok, L} = lua:new_state(),

  % put "print" global on top of stack
  lua:getfield(L, global, "print"),

  % put hello on top
  lua:pushstring(L, "Hello from Lua!"),

  % execute on top 2 values on stack
  lua:call(L, 1, 0).
```



# Erlualib



hello() ->

```
{ok, L} = lua:new_state(),  
lua:getfield(L, global, "print"),  
lua:pushstring(L, "Hello from Lua!"),  
lua:call(L, 1, 0).
```





# Luerl



- New Lua VM programmed *in* Erlang
- Perfectly secure way
- Precompiled Chunks
- Re-usable State
- Work in progress



# Luerl



```
hello() ->
```

```
L = luerl:init(),
```

```
luerl:do("print('Hello from Lua!')", L).
```



# Luerl



hello() ->

```
L = luerl:init(),  
  Chunk = luerl:eval("print('Hello from Lua!')", L),  
  luerl:call(Chunk, L)
```

% Chunk and L can be re-used.



# Benchmark

Luerl vs Erlualib



Erlualib / PUC

Luerl

270  $\mu$ s

state Init

250  $\mu$ s

10  $\mu$ s

parse 1 + 1

15  $\mu$ s

2.5  $\mu$ s

execute 1 + 1

13  $\mu$ s

23  $\mu$ s

parse formula

63  $\mu$ s

2.5  $\mu$ s

execute formula

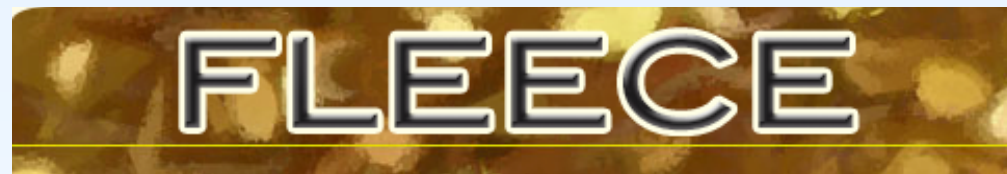
35  $\mu$ s

2.5  $\mu$ s

call a function

28  $\mu$ s

Formula:  $a = 7.33$ ;  $b = 9000$ ;  $c = (33 * a / b) ^ 15 * a + b$



**Fast Lua to JSON encoder**

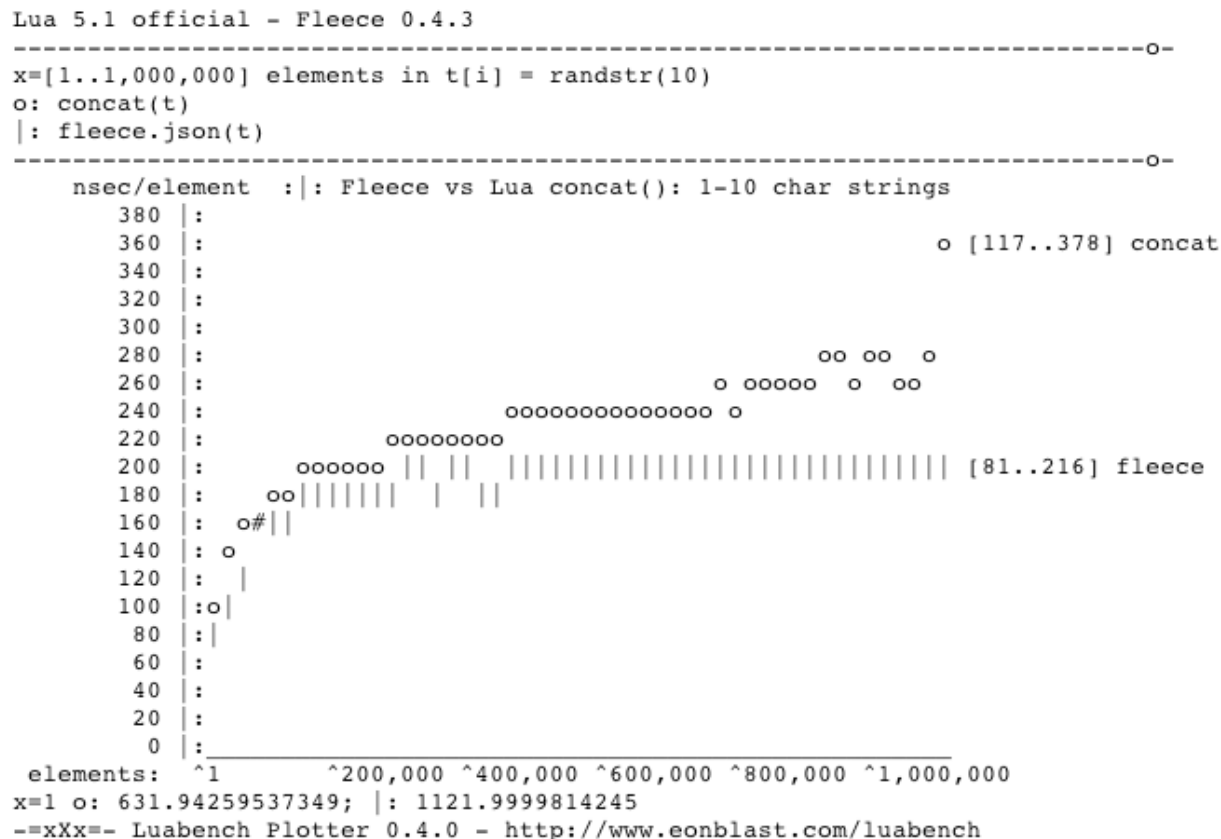


# Fleece

- Fastest Lua JSON encoder
- C+ASM or ISO C
- Faster than Lua JIT
- Faster than concat
- 10x faster than other fastest

## Graphs

The bench10\* scripts, among others, produce pretty graphs, like the one shown below. Find a list of benchmark result pages on page [doc/BENCHMARKS.html](http://www.eonblast.com/doc/BENCHMARKS.html).





# Erlvolt



- Native Erlang VoltDB driver
- Async parallel data shuffling
- Coming next quarter
- Your help is wanted
- Connect over github



# Server Architecture

- Erlang
- Lua
- JSON
- VoltDB
- Luerl
- Erlualib
- Fleece
- Erlvolt





# Invitation

- All at github
- All w/samples & docs
- All maintained by Eonblast
- Luerl maintained by R. Virding



Updates @hdiedrich, @rvirding

# Questions

Mail: [hdiedrich @ eonblast.com](mailto:hdiedrich@eonblast.com)

Skype: eonblast

Twitter: @hdiedrich