

# Caching in an Erlang HTTP stack

Enrique Paz @quiquepaz





- Software Architect
- Passionate Erlang developer
- Testing enthusiast
- Love beautiful code!



Gamesgames.com



guest

JOIN FOR FREE or

Sign In



Play Free Online Games on GamesGames.com

Search for Games

Search for...

HOME

ACTION

ADVENTURE

BOARD &amp; CARD

GIRLS

MULTIPLAYER

PUZZLE

RACING

SKILL

MORE



Follow

## GAMES

New Games

Popular Games

My Favorites

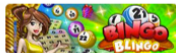
Social Games

Mobile games

Videos

Cash Games (King)

Mundigames



Animal Games

Action

Adventure

Basketball

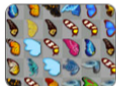
Beauty

Match 3

## RECOMMENDATIONS



Family Barn



Butterfly Kyodal



Hair Expert



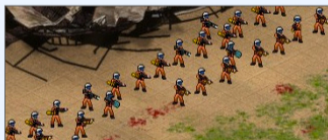
Slotomania



Operate Now: Nose...



Traffic Slam 3



## Desert Moon

On a distant world, a ragtag group of misfits fights for survival against evil from beyond the stars.

1

2

3

PLAY NOW

## NEW GAMES



ADVERTISEMENT

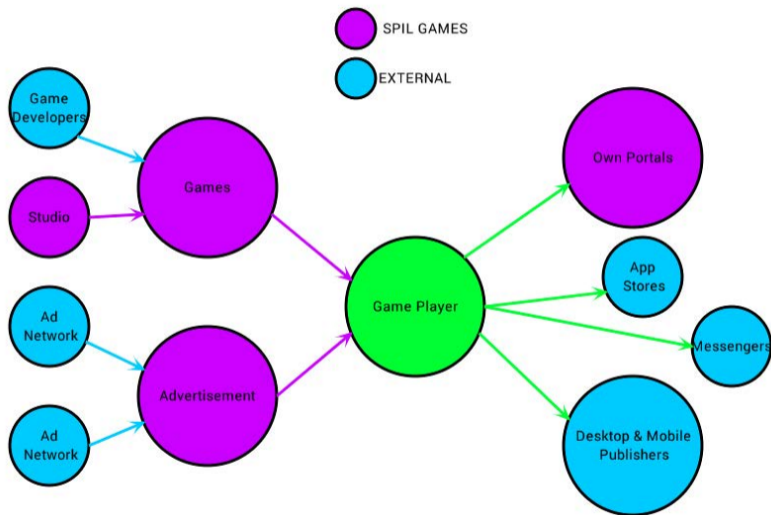
10.- korting voor nieuwe klanten  
waardeboncode: 'NIEUW'

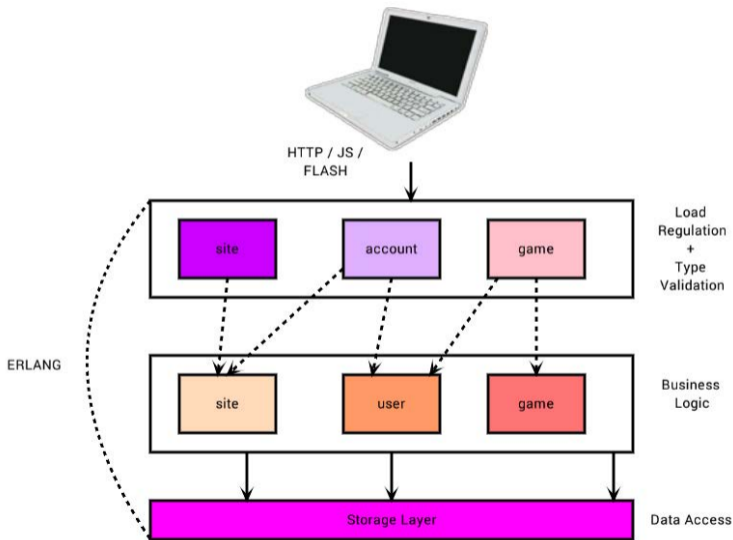


eerst wehkamp.nl

## POPULAR







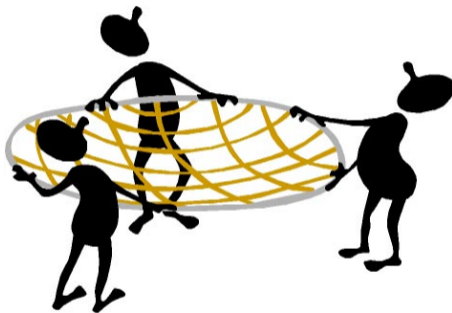
# Why Caching?



- To avoid disk/DB access
- To avoid calculations
- To avoid resource bottlenecks



- To reduce backend load
- To mitigate outages impact



**Careful!**

**Do Not Depend On Cache!**



## Common pitfalls

- Caching layers
- Deployments and cold start misses
- Server maintenance (on caching pools)
- Cache flushing for debugging

- Dataset & Entry size
- Should it survive restart?
  - ▶ Distribution VS replication

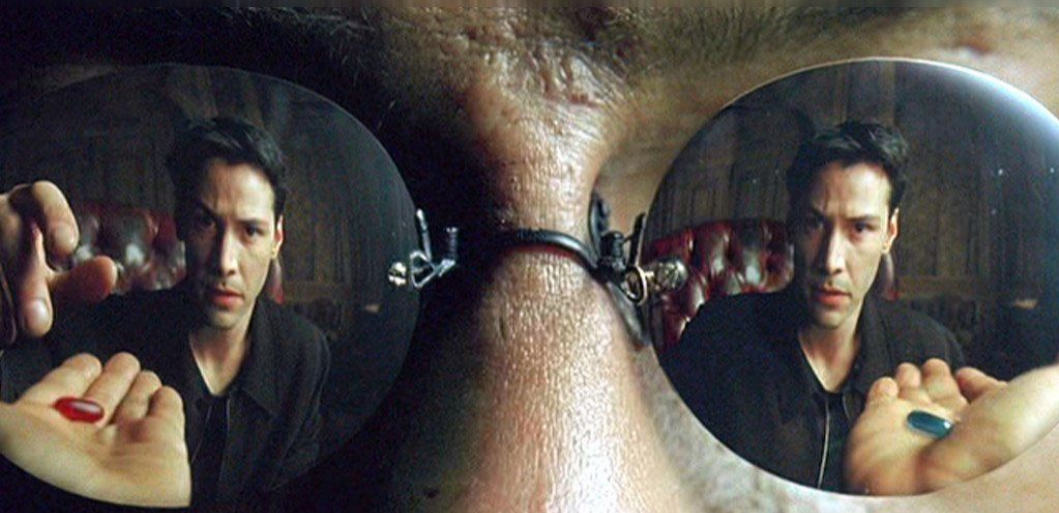


- Amount of different keys
- Request & Update Frequency
  - ▶ Validity & eviction strategies



- Caching Error Values
  - ▶ If done, how to recover quickly?
  - ▶ If not done, what about overload?





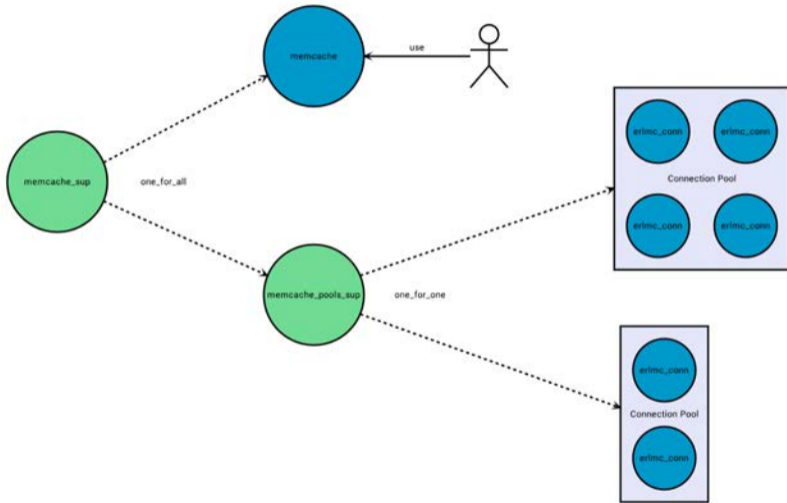
- Not reinventing the wheel
- Distributed caching
- Caching outside the node
- System Administrators



## erl-memcache

- Memcached
- Pool of tcp connections
- Memcached binary protocol
- Detailed memory management
- If local, memcached can be supervised





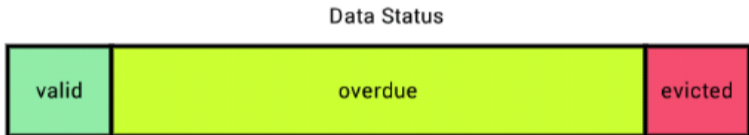
- Minimizing disk/DB access in Spilgames Storage Platform
- <https://github.com/spilgames/erl-memcache>

## erl-cache

- In node caching (small datasets)
- Concurrency + long lived data
- Per key caching strategies
- Error awareness
- Non intrusive memoization



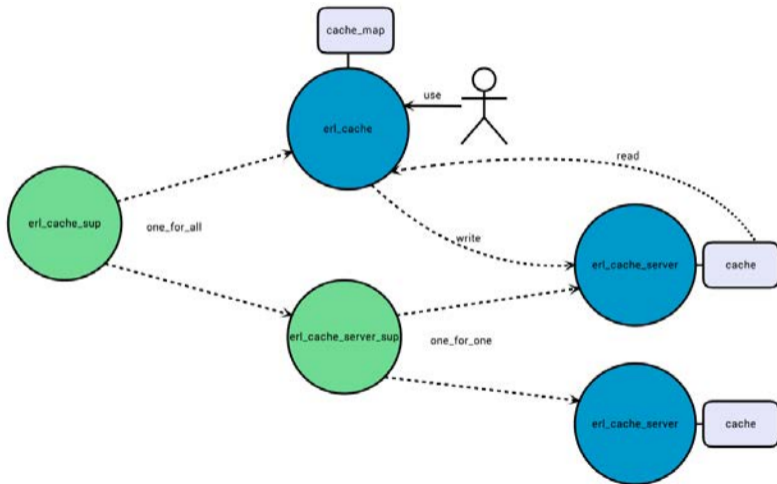
- gen\_servers with protected ETSs
- Per cache server stats
- Periodic eviction
- Auto refresh overdue entries



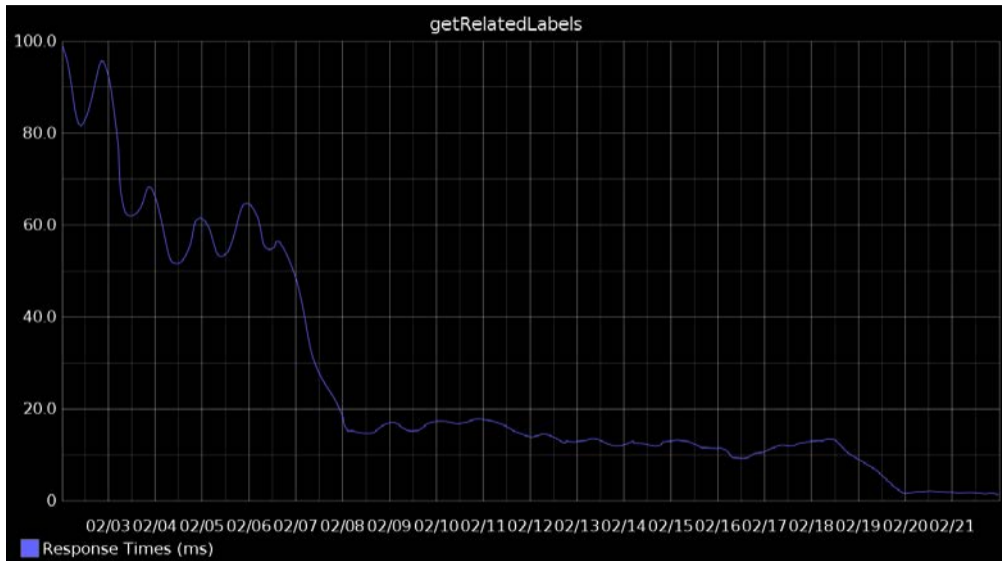
- start\_cache(Cache, DefaultOpts)
- stop\_cache(Cache)
- get(Cache, Key, Opts)
- set(Cache, Key, Value, Opts)
- evict(Cache, Key)
- get\_stats(Cache)

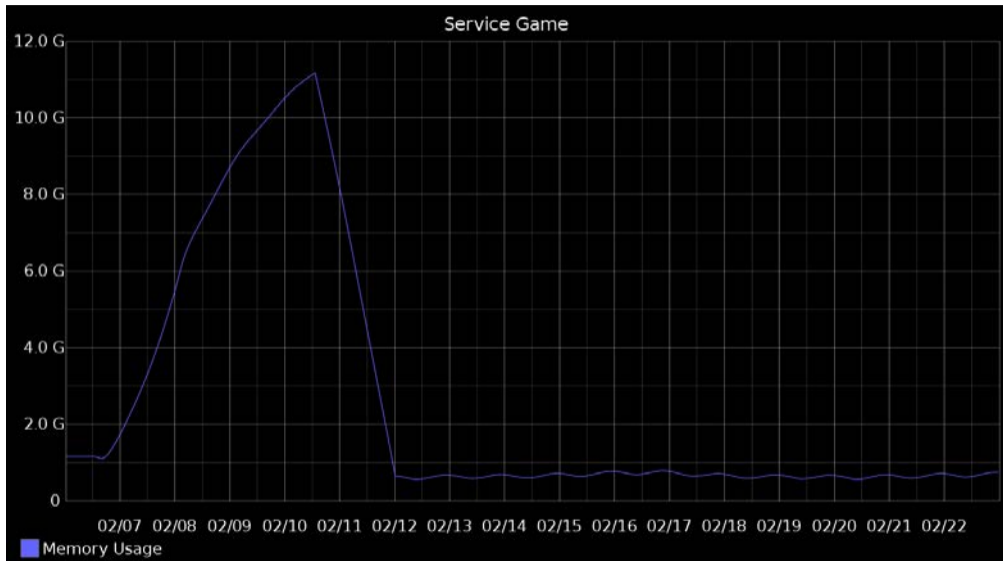
```
?CACHE(s1, [{wait_for_refresh, false}]).  
operate(Input) ->  
    {ok, Output} = extra_complex_op(Input),  
    Output.
```

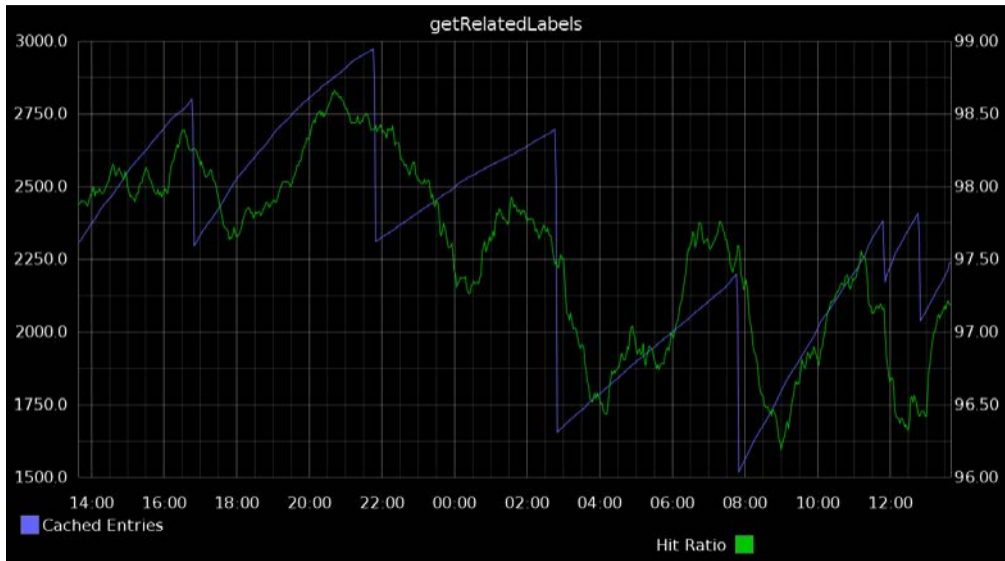




- Avoiding inline expensive calculations
- Protecting the app running the bussiness logic
- <https://github.com/spilgames/erl-cache>



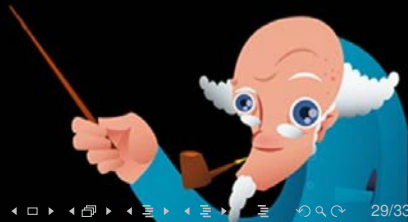




# Today's Lessons

Memory Control  
+ Survive Restarts  
Distributed Caching

-----  
erl-memcache

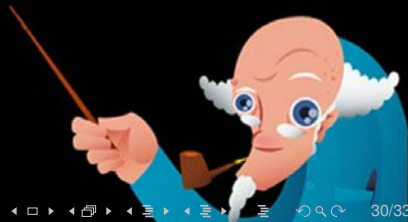


# Today's Lessons

Small Data Set  
+ Auto refresh  
Error Awareness

-----

erl-cache



- **Poolboy** for worker pools
- **Lager** for logging

- <http://github.com/spilgames/erl-memcache>
- <http://github.com/spilgames/erl-cache>
- <http://github.com/spilgames/erl-decorator-pt>

