Damian Dobroczyński, mFinder sp. z o.o.

qoocku at after f but before h mail dot com

qoocku at m dash finder dot pl

http://www.mfinder.pl

Quite Optimistic Story about

mFinder Server

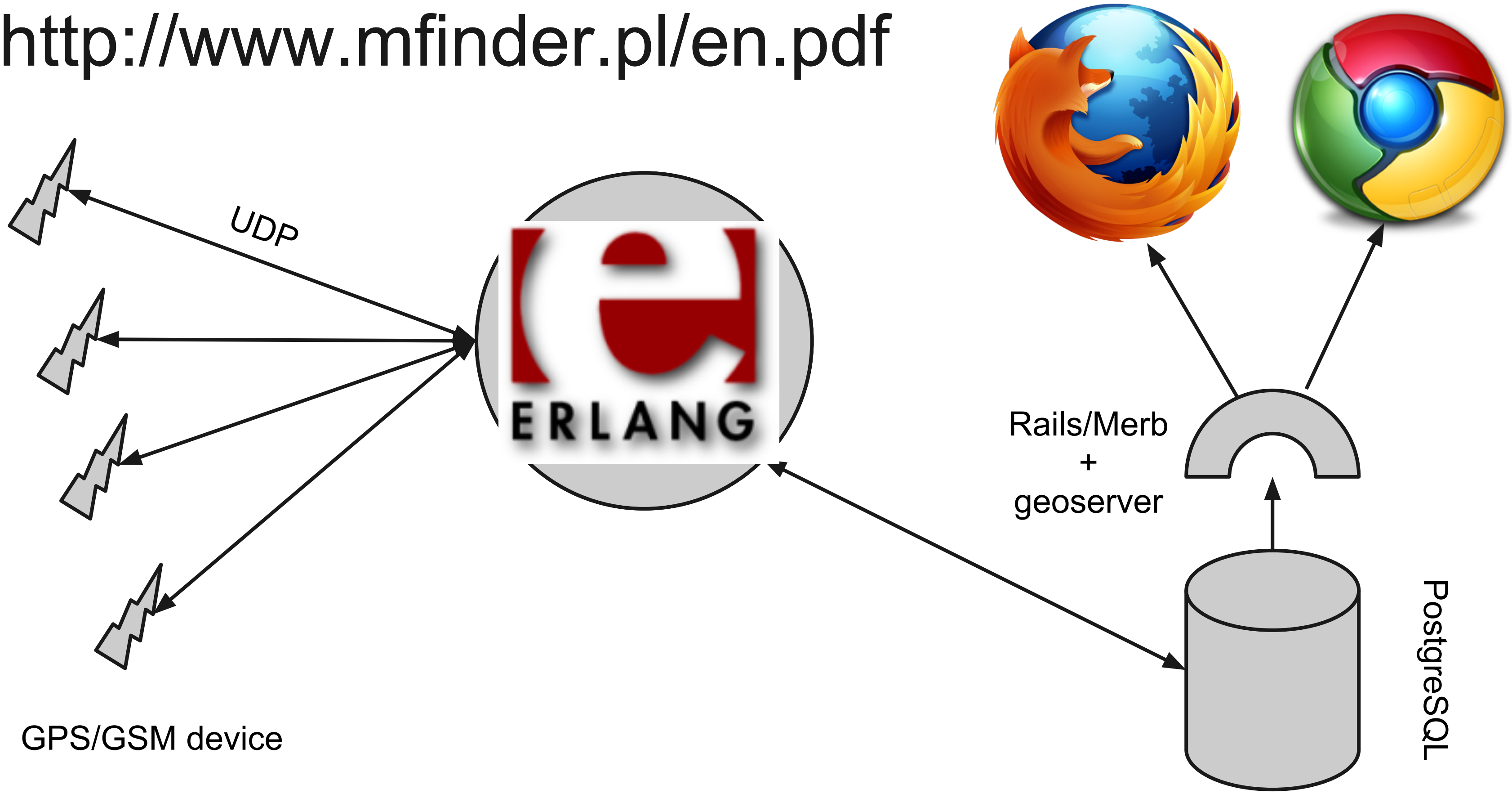
a server for city transport fleet tracking system

Speaker's background

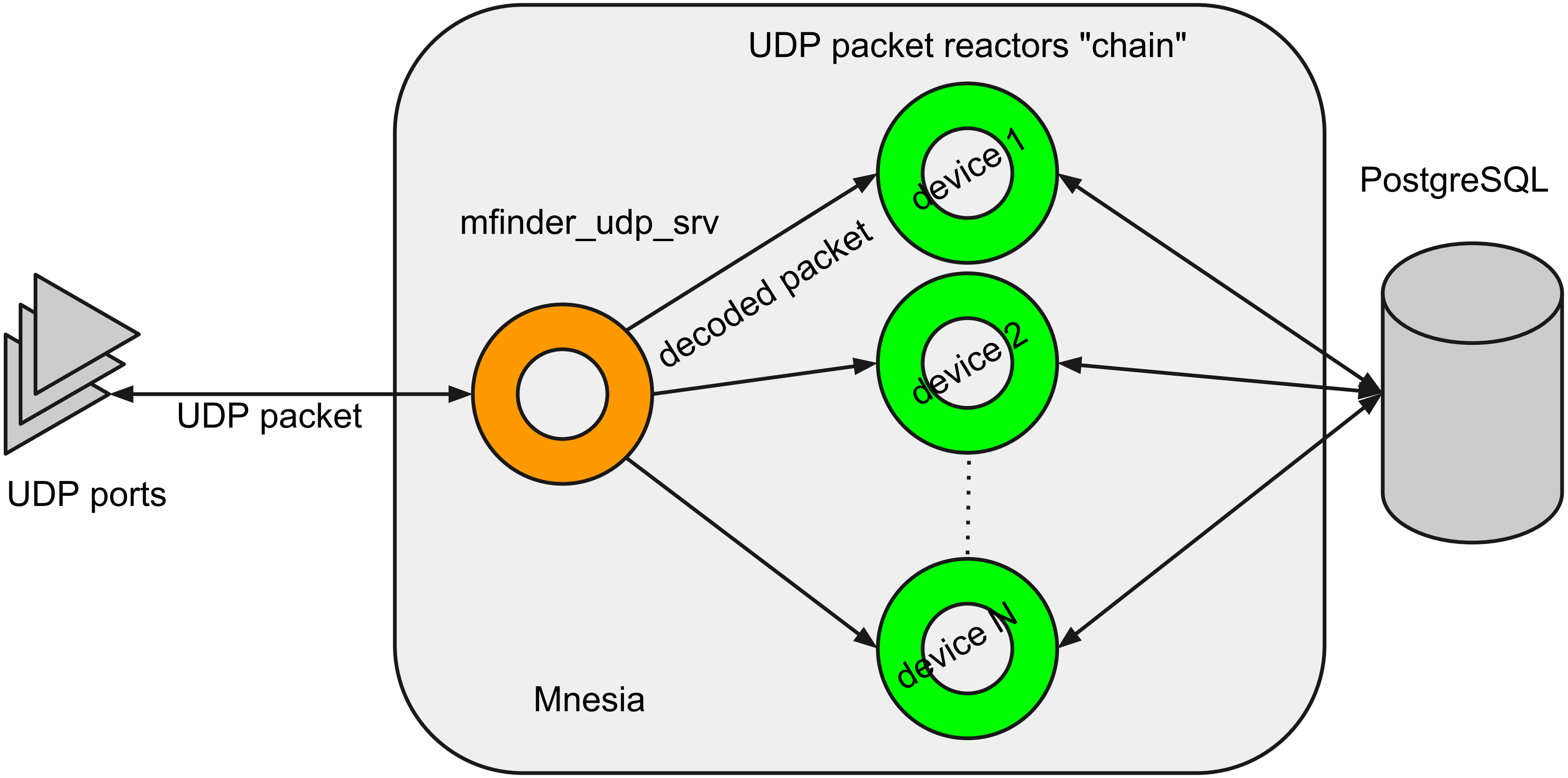
- education: automation & robotics
- die-hard C++/Python user & programmer
- 2005: first look at Erlang - some stuff related to XMPP servers. No "classes", no worth using.
- 2008: an inspiring blog entry found (http://blog.zabiello.com/2008/02/29/erlang) - "Who's gonna face Erlang?"
- 2008/2009: J. Armstrong's book + K. Smith's screencasts (days when dollar was cheap)

mFinder System - intro

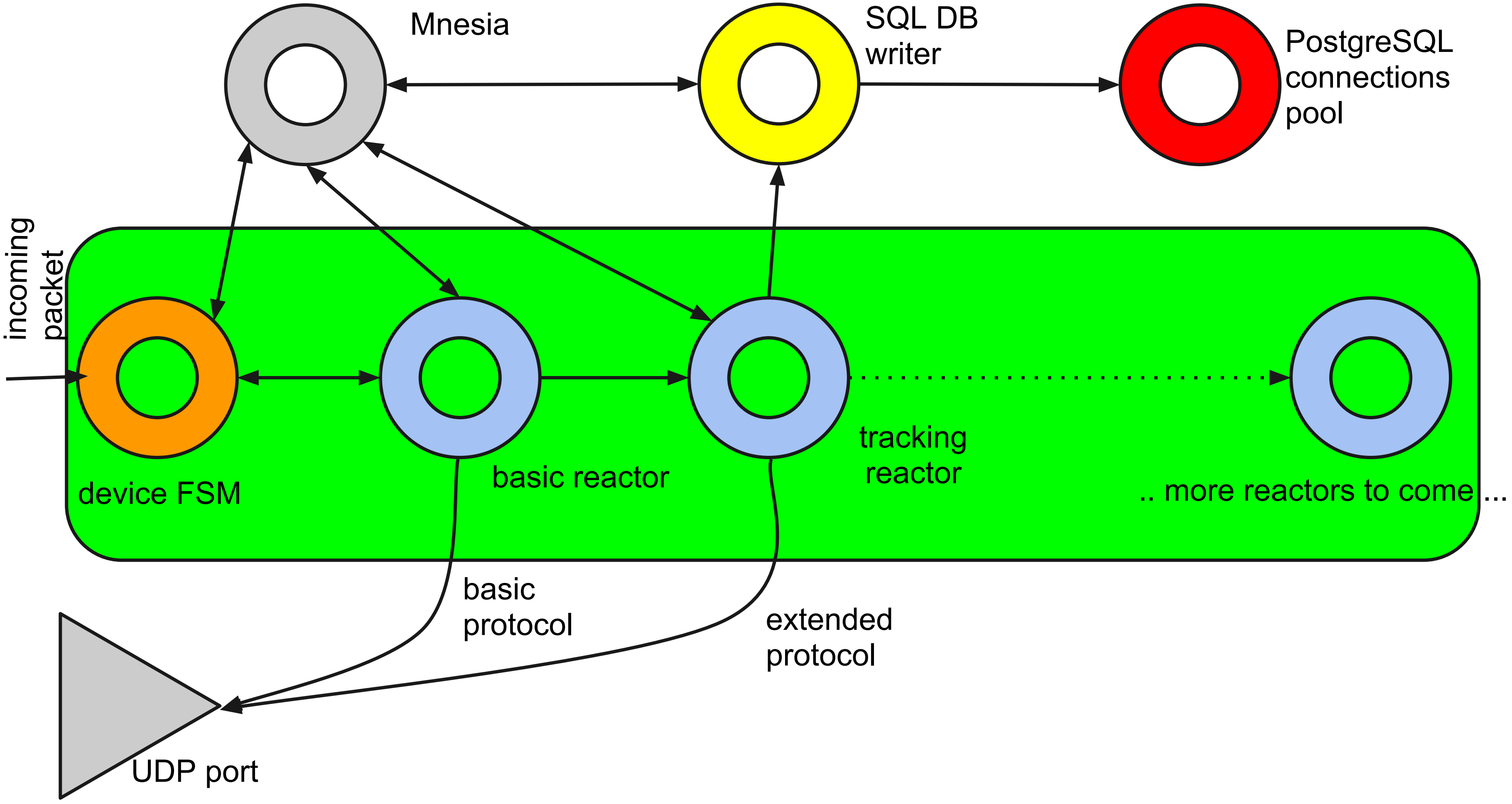
http://www.mfinder.pl/en.pdf



mFinder server - general view



mFinder server - reactors



Basic Reactor

- validates incoming GPS position packets (relative to a reference point or absolute)
- generates global position message for next reactors - sends it further
- stores current global position (Mnesia)
- sends "scaling" packet to the device (new reference position)
- generates events if event packet has come (engine events, emergency lantern state, passenger doors etc.)

Tracking Reactor

- reacts on packets related to planned departures schedule
- copies specific departure timetables from PostgreSQL to Mnesia
- tries to find current delay or overspeed based on departure timetables and current global position
- generates data for SQL DB writer containing current vehicle state (delay, approached/left stops etc.)

mFinder server - summary

- 31 core modules, a dozen of auxiliary modules
- OTP compliant processes
- (4 persistent basic processes + several transient) / device
- some tests with simulated fleet
- officially presented on TRANSEXPO 2009
- in production (since 2010)
- earns for itself - no golden eggs though
- mTransVIP project to finish

Erlang after mFinder - a few thoughts (1)

- Process supervising works so well - to my surprise!
- Fear to upgrade a release in OTP-ish manner: using incremental code change and manual code hot-swapping instead (no code_change/3 usage!).
- Few downtimes caused by PostgreSQL failures and my ad-hoc upgrade experiments.

Erlang after mFinder - a few thoughts (2)

- dialyzer is overwhelming and its messages seem to be cryptic - still not sure when to use it and how much should I care?
- unified standard "collections" API would be just fine
- a temptation to re-use parts of implementation on the functional side (extending/mixing other modules)
- still trying to implement "polimorphic" actors - ex-objectoholic's customs are strong

And this is...

... the end of this story.