COFFEEBEAM
A BEAM VM FOR ANDROID

Viktor Gergely
Software Developer / Tech Lead @ Erlang Solutions

@github vikger/coffeebeam
viktor.gergely@erlang-solutions.com
QUESTIONS TO BE ANSWERED

What is possible?
What do you need to make it happen?
How do the parts fit together?
How can the VM interact with your phone?
Using Erlang to build Android applications that are

- Robust  OK
- Concurrent  OK
- Scalable  well...
- Soft real-time  hmm...
IS THIS POSSIBLE?

Tic Tac Toe
Computer's turn

NEW GAME

XOX
XOX
OXX
WHAT YOU NEED

▸ Make Android understand BEAM
▸ Load BEAM in Android way
▸ Create a VM that executes the BEAM
▸ Make the VM interact with Android

Android requires a different approach
Build a VM with high-level structures
Low CPU, high memory usage
HOW THE PARTS FIT TOGETHER

VM (Java) ➔ .jar ➔ Activity ➔ .apk

BEAM files ➔ BeamClient

How to build your Android application with Erlang VM and modules
INTERACTION BETWEEN BEAM AND PHONE

Start VM
Stop VM
Load BEAM file
Apply function

BeamClient

YourOwnBeamClient

Handle function result
Handle function call
COMMUNICATION

Android
- Start activity -> apply(m, f, a)
- handleResult(GamePid)

New Game button -> apply(m, f, a)
- handleCall(update, {new_game, Board})

Click (x, y) -> apply(m, f, a)
- handleCall(update, {put_player, Board})
- handleCall(update, {put_computer, Board})
- handleCall(update, {draw, Board})

CoffeeBeam VM
- tictactoe:start() -> GamePid

- tictactoe:new_game(GamePid)
- beamclient:update({new_game, Board})

- tictactoe:put(GamePid, X, Y)
- beamclient:update({put_player, Board})
- beamclient:update({put_computer, Board})

- handleCall(update, {put_player, Board})
- handleCall(update, {put_computer, Board})
- handleCall(update, {draw, Board})
- beamclient:update({draw, Board})
CONTRIBUTION

▸ Public project:
https://github.com/vikger/coffeebeam

▸ Future plans
  ▶ IO
  ▶ Network support
  ▶ Node management
  ▶ Advanced number handling
  ▶ Lots of BIFs
USEFUL READING

▸ The BEAM Book: https://blog.stenmans.org/theBeamBook
▸ BEAM VM Wisdoms: http://beam-wisdoms.clau.se/
▸ External term format: http://erlang.org/doc/apps/erts/erl_ext_dist.html
THANK YOU
Q&A

Viktor Gergely
Software Developer / Tech Lead @ Erlang Solutions

@github vikger/coffeebeam
viktor.gergely@erlang-solutions.com
STUDY OF EXISTING SOLUTIONS

▸ Building Erlang for Android
  ▸ [https://bluishcoder.co.nz/2015/06/21/building-erlang-for-android.html](https://bluishcoder.co.nz/2015/06/21/building-erlang-for-android.html)
  ▸ Complicated configuration and installation for non-rooted devices

▸ Erlang4Android
  ▸ [https://code.google.com/archive/p/erlang4android/](https://code.google.com/archive/p/erlang4android/)
  ▸ Depends on Scripting Layer for Android (SL4A)
  ▸ May not be suitable for custom Android apps

▸ Erjang
  ▸ [https://github.com/trifork/erjang/issues/63](https://github.com/trifork/erjang/issues/63)
  ▸ Involves code recompilation differently from Android way
  ▸ Android application not on horizon
THE VM IN AN ANDROID ACTIVITY

onCreate()
onStart()
onStop()
onDestroy()

Erlang VM lifetime

Unsafe

The activity may be stopped when hidden
TYPES MATTER!

ErlTerm

parse()
toString()

ErlNumber

ErlInt
ErlFloat
ErlAtom

...
THE VM STRUCTURE

- Track loaded modules
- One scheduler for each VM
- Logger
- Registered processes
- Provide interface to scheduler
  - Spawn new process
  - Send
  - Timeouts