Pilgrim's Progress to the Promised Land

Robert Virding

Erlang SOLUTIONS

www.erlang-solutions.com

The Pilgrimage to the Promised Land

You have heard the Prophets and their followers extolling Elixir as the true path to the Promised Land of concurrent, fault-tolerant and scalable systems and have decided to make the pilgrimage.

Are You Following the Right Prophets?

Are You Following the Right Prophets?

Have you chosen the right path?

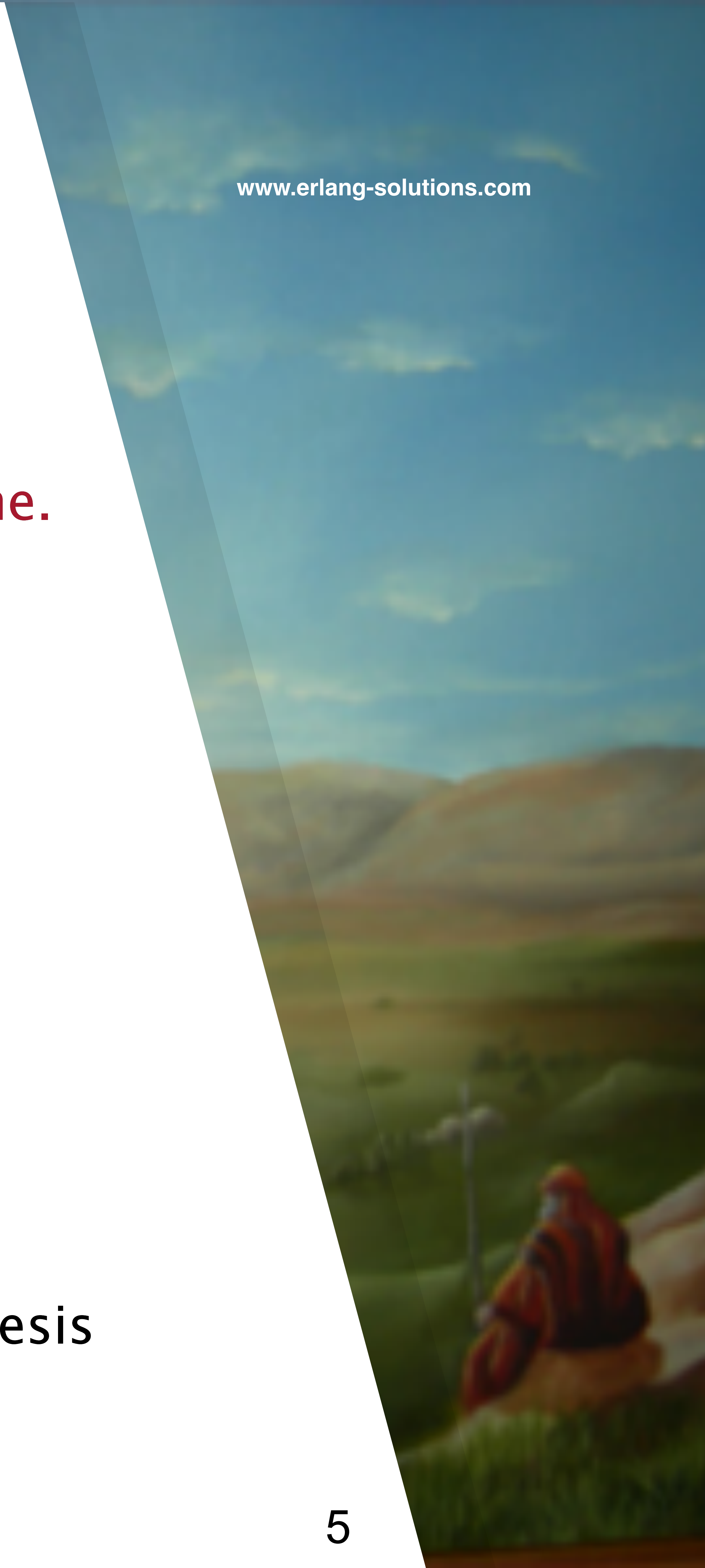
Is Elixir the right language/system for you?

Are You Following the Right Prophets?

- Handle a very large numbers of concurrent activities.
- Actions must be performed at a certain point in time or within a certain time.
- System distributed over several computers.
- Interaction with hardware.
- Very large software systems.
- Complex functionality such as feature interaction.
- Continuous operation over many years.
- Software maintenance (reconfiguration etc.) without stopping the system.
- Stringent quality and reliability requirements.
- Fault tolerance both to hardware failures and software errors.

Not just telecom

Bjarne Däcker, November 2000 - Licentiate Thesis



Are You Following the Right Prophets?

Erlang/OTP, and hence Elixir, are designed for systems with properties:

- Lightweight, massive concurrency
- Fault-tolerance must be provided
- Timing constraints
- Continuous operation for a long time
- Continuous maintenance/evolution of the system
- Distributed systems

Are You Following the Right Prophets?

Erlang/OTP and Elixir are not good for systems with:

- Heavy number crunching
- Global, shared, mutable data
- ...

BUT

we are good at interacting with other systems

The Joy of the First Experiment

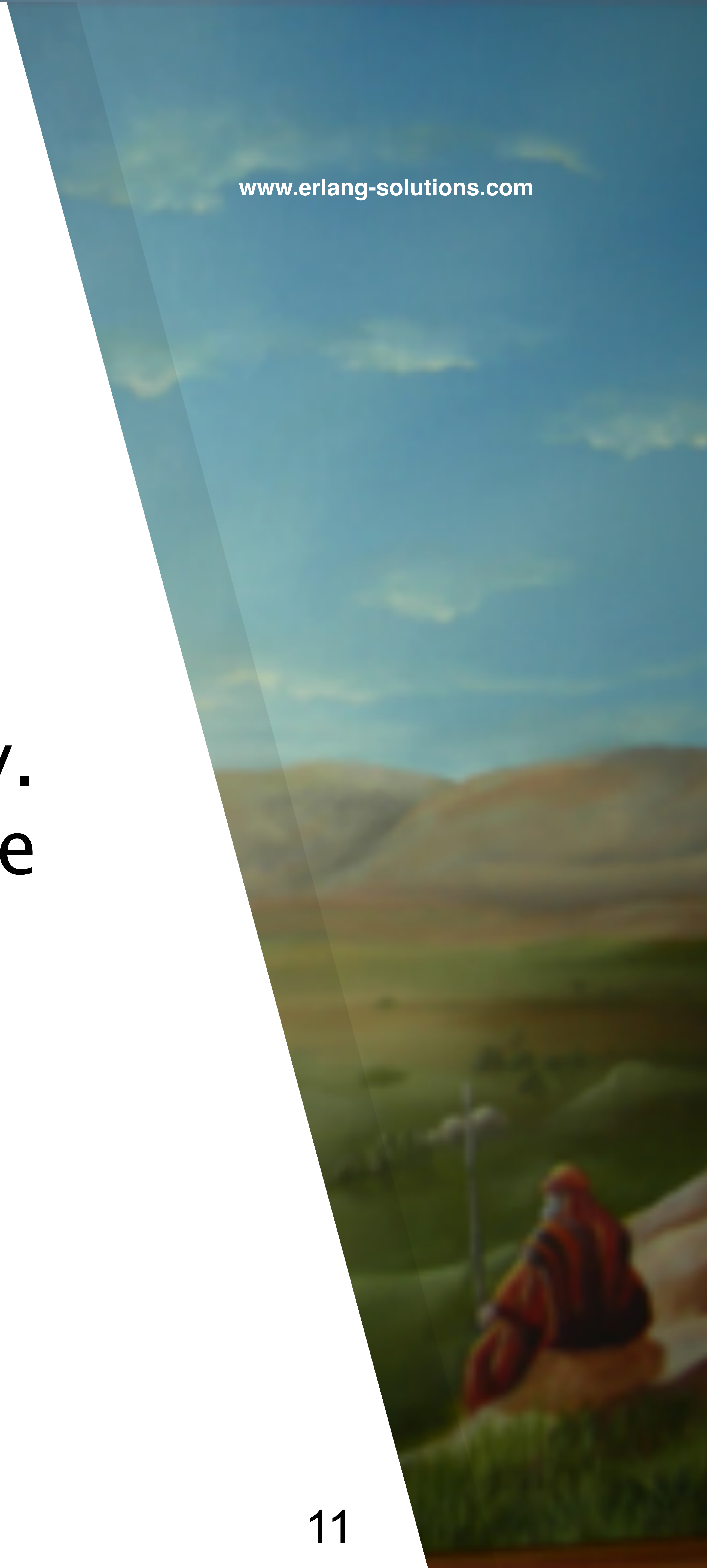
The Joy of the First Experiment

You have downloaded the system, used the tools and have written your first simple Elixir server or Phoenix webpage, and it works.

The Village of Confusion

The Village of Confusion

Yes, your system is working but behaving strangely.
Did you really listen to the prophets when they were describing the language/system?



The Village of Confusion

- Elixir may **look** like Ruby, but it is not Ruby
- Elixir is not OO
- Elixir has "strange" handling of data
- Elixir has "strange" way of building systems

The Hill of Functional

The Hill of Functional

There is now a hill we must climb to gain understanding of the language. It is not high.

The Hill of Functional

Elixir is a functional language with:

- Immutable data
- Pattern matching
- Recursion
- ...

The Mountain of Concurrency

The Mountain of Concurrency

There is now a mountain we must climb to start gaining understanding of building systems. It is not high though the path may feel different.

The Mountain of Concurrency

Elixir has a different way of structuring systems:

- Isolated processes
- No sharing of data
- No global data
- Communication through asynchronous messages
- We aren't scared of creating lots of processes
- We aren't scared of crashing things

The Marsh of Fault-Tolerant Systems

The Marsh of Fault-Tolerant Systems

There is a big marsh which we need to make our way through by building a fault-tolerant system. There are many paths but we need to pick the best one

The Marsh of Fault-Tolerant Systems

The path of **OTP** is the obvious choice:

- It was designed to support building fault-tolerant systems
- Many of things we have probably already used are part of OTP
- The underlying system is built on it.
- Elixir already extensively uses it and has extended it
- OTP is extensible when necessary

BUT

It does require thinking about how your should behave **WHEN** things go wrong

The Double-Edged Sword of Metaprogramming

www.erlang-solutions.com

The Double-Edged Sword of Metaprogramming

You have been carrying the Sword of Metaprogramming and already used its effects without realising its power. It has the power to do both good, and EVIL.

The Double-Edged Sword of Metaprogramming

Macros are a powerful tool which lets you build code at runtime

- Used wisely they can really be helpful
- Used unwisely they can lead to totally incomprehensible code
- You are already using macros:
  - def and defp

The Language Heaven is Multilingual

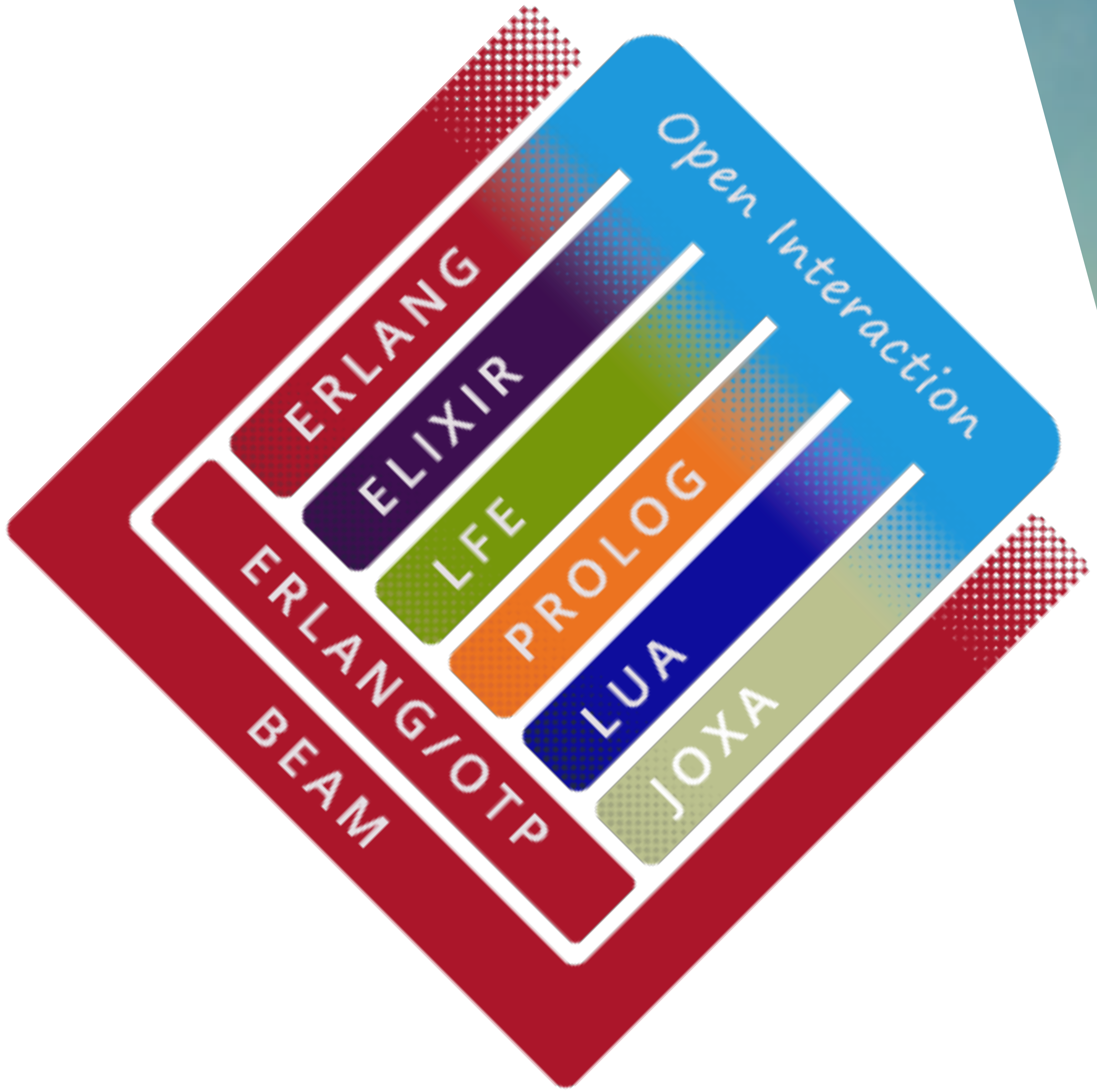
The Language Heaven is Multilingual

On your travels you have probably heard tales that the Language Heaven of the Promised Land is multilingual and that there is squabbling between the language gods and between their followers.

DON'T DESPAIR!

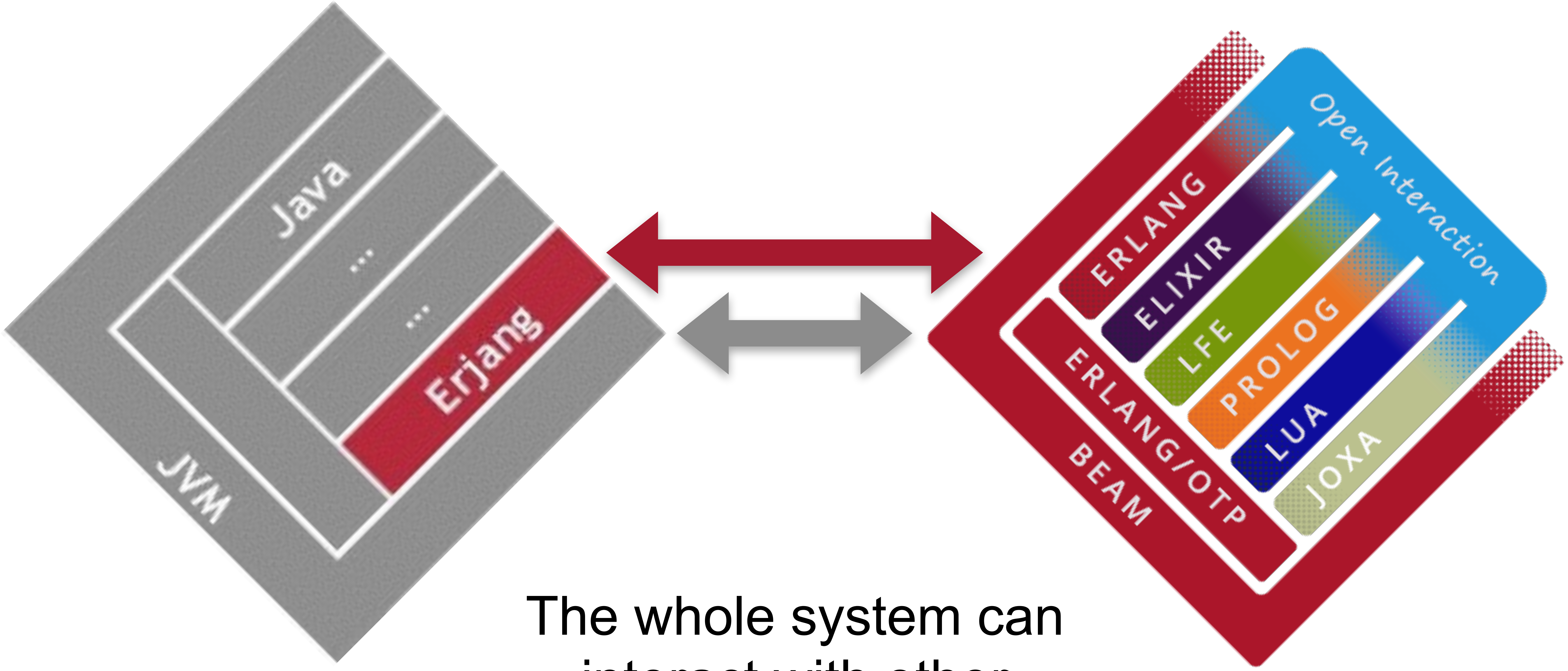
The Language Heaven is Multilingual

The Language Heaven of the Promised Land is really the ERLANG ECOSYSTEM with all the languages built/running on top of the BEAM, Erlang and OTP.



By following "the rules" the languages openly interact with, and support, each other making the whole system more powerful than any individual language can ever be

The Language Heaven is Multilingual



The whole system can interact with other systems

www.erlang-solutions.com

The Pool of Introspection

The Pool of Introspection

Sit by the pool and in the reflections of its waters look inside your system while it runs, introspect and gain understanding.

The Pool of Introspection

There are a number of tools that allow you to introspect running systems:

- Observer – part of the standard release
- Dbg – part of the standard release
- Recon – common 3rd party
- Redbug – common 3rd party
- WombatOAM – commercial product

These work in Erlang but the mapping is very straightforward.

The Joy of Arrival

The Joy of Arrival

You have built your first REAL system and it works. It can handle all the load, survive errors and just keep going.

Rejoice, and realise the path wasn't really as difficult as it at first seemed.

www.erlang-solutions.com

Robert Virding
rvirding@gmail.com
robert.virding@erlang-solutions.com
@rvirding