HELLO ELIXIR!

Sameeri Marryboyina

THANK YOU SPONSORS

Premier Sponsors

amazon web services

WORK READY OKLAHOMA

Gold Sponsors

mozilla

GABLE GOTWALS COUNSEL

NUTANIX

ROAD MAP TECHNOLOGIES

J.B. HUNT

Silver Sponsors

ASPOSE Your File Format APIs

Lunch Sponsors

Microsoft

Resource Sponsors

isocentric networks
internet solutions advancing business

Prize Sponsors

ASPOSE Your File Format APIs

CODESMITH Tools

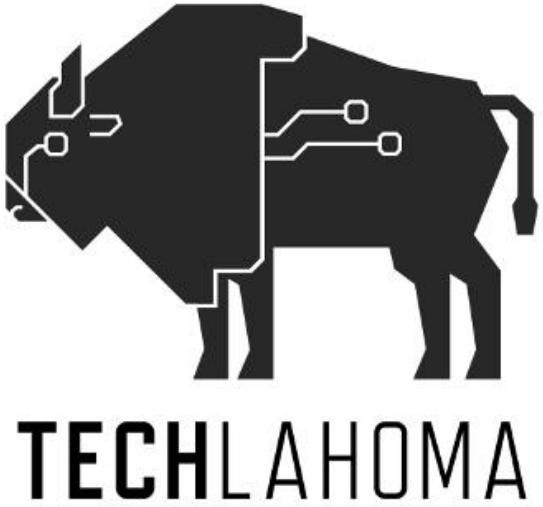
LogicNP Software
The power of Components

McALISTER'S DELI

Bounce U

THANK YOU TECHLAHOMA!

OKC FP



OKC Elixir

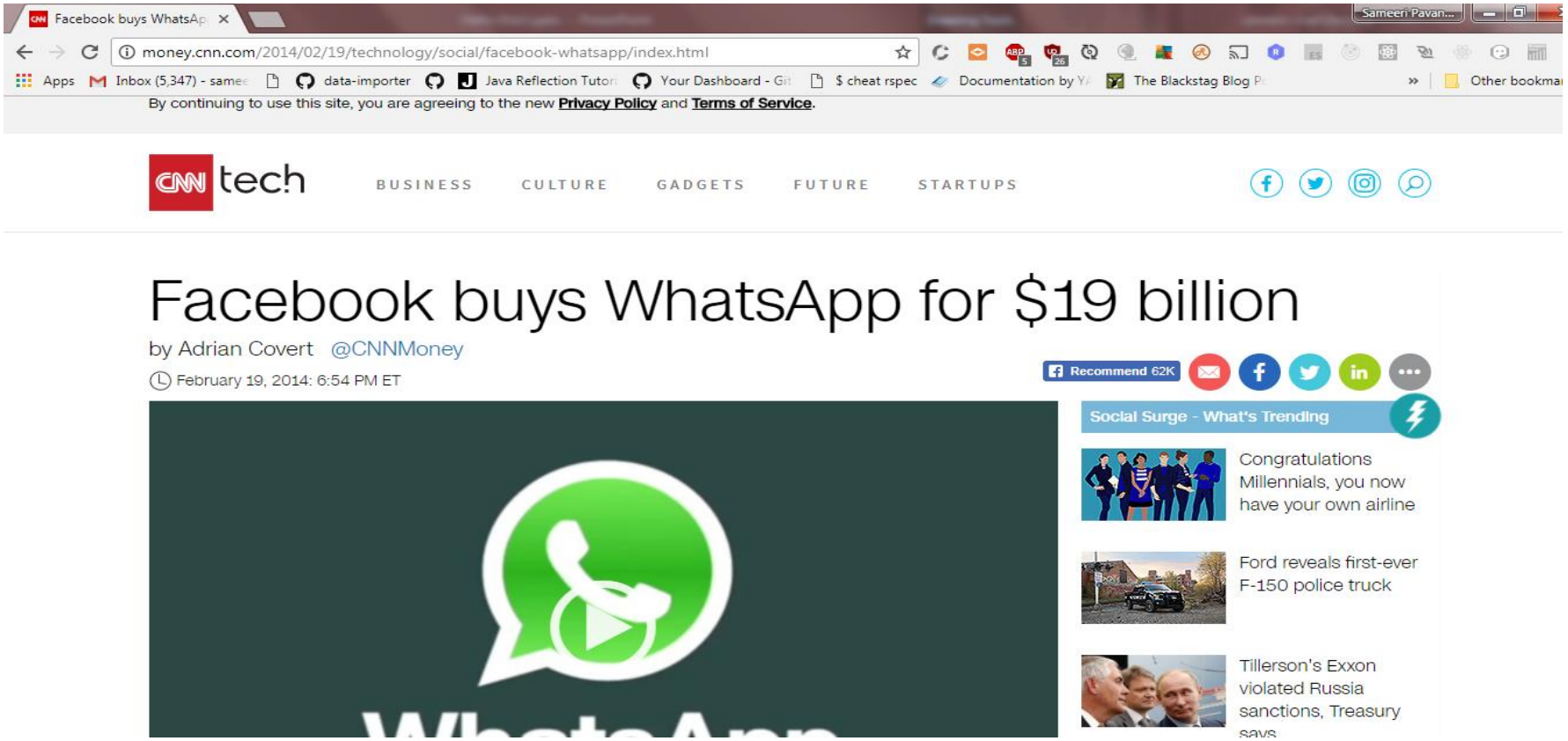
Find Techlahoma on all things social.

TALK AGENDA

- Target Audience
- About Me
- Foundations
- The road to Elixir (How I met Elixir!)
- Elixir

TARGET AUDIENCE

TALK PITCH


Facebook buys WhatsAp
money.cnn.com/2014/02/19/technology/social/facebook-whatsapp/index.html
Apps
Inbox (5,347) - same
data-importer
Java Reflection Tutor
Your Dashboard - Git
$ cheat rspec
Documentation by YA
The Blackstag Blog P
Other bookma
By continuing to use this site, you are agreeing to the new Privacy Policy and Terms of Service.
CNN tech
BUSINESS
CULTURE
GADGETS
FUTURE
STARTUPS
Facebook buys WhatsApp for $19 billion
by Adrian Covert @CNNMoney
February 19, 2014: 6:54 PM ET
Recommend 62K
Social Surge - What's Trending
Congratulations Millennials, you now have your own airline
Ford reveals first-ever F-150 police truck
Tillerson's Exxon violated Russia sanctions, Treasury

THE WHATSAPP STORY

- WhatsApp, was incorporated in 2009 by Brian Acton and Jan Koum, both former employees of *Yahoo!* . After Koum and Acton left Yahoo! in September 2007, the duo traveled to South America as a break from work.[18] At one point they applied for jobs at Facebook but were rejected.[18] For the rest of the following years Koum relied on his $400,000 savings from Yahoo!. In January 2009, after purchasing an iPhone and realizing that the App Store would soon create an industry of apps, Koum started visiting his friend Alex Fishman in West San Jose where the three would discuss ".... having statuses next to individual names of the people", but this was not possible without an iPhone developer. Fishman found a Russian developer on RentACoder.com, Igor Solomennikov, and introduced him to Koum. Koum named the app "WhatsApp" to sound like "what's up". On February 24, 2009, he incorporated WhatsApp Inc. in California. However, because early versions of WhatsApp often crashed or got stuck at a particular point, Koum felt like giving up and looking for a new job, upon which Acton encouraged him to wait for a "few more months". - Wikipedia

WHAT THIS TALK IS NOT!

- This talk is not entirely about how to code in Elixir!

WHAT THIS TALK IS ABOUT

- “This talk is a story!”

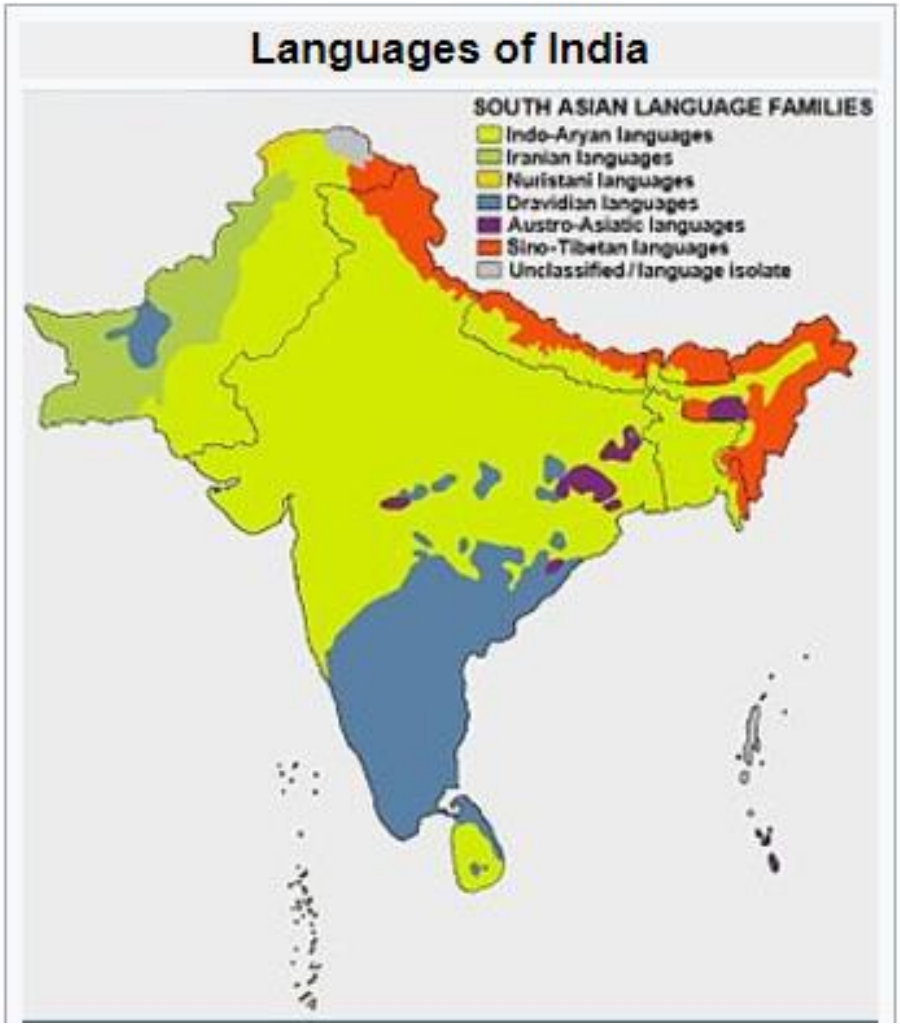
ABOUT ME

HELLO, SAMEERI.

- Sameeri Pavan Kumar Marryboyina
- Hyderabad, India
- I can communicate in Telugu, Hindi, English, German(very little)
- MS in Computer Science, The University of Oklahoma
- Senior Software Engineer, Telogical Systems LLC
- Full Stack JS - React, Redux and its Ecosystem, Node, Express
- Norman, OK

INDIA

- Let us travel to Sameeri's home..
- https://www.google.com/maps/place/India/@18.2937219,71.4458457,5z/data=!4m5!3m4!1s0x30635ff06b92b791:0xd78c4fa1854213a6!8m2!3d20.593684!4d78.96288



Language families of the Indian subcontinent; Nihali, Kusunda, and Thai languages are not shown.

Official languages	Assamese Bengali Bodo Dogri English[1][2][3] Gujarati Hindi[1][4] Kannada Kashmiri Konkani Maithili Malayalam Marathi Meitei (Manipuri) Nepali Odia Punjabi Sanskrit Santali Sindhi Tamil Telugu Urdu (total: 23, including 22 8th Schedule languages and additional official language, English)
Main foreign languages	• English – around 125 million speakers[5] • French – approximately 75,000 speakers

source : https://en.wikipedia.org/wiki/Languages_of_India

FAMILY

- Dad – Our German and English Teacher
- Brother – German, Spanish. Interested in Japanese, Italian, Korean
- Mom – Was a Telugu lecturer at a college.

FOUNDATIONS

BEING A POLYGLOT...

WORK	SCHOOL	LEARNING	RADAR
JavaScript	C	Elixir	Elm
C#	C++/VC++	Erlang	TypeScript
Ruby	COBOL		Python
Java	Assembly		Scala
			Rust
			Go

BUT, WHAT DOES IT MEAN TO BE A LANGUAGE?

WHAT KIND OF A PROBLEM ARE WE TRYING TO SOLVE?

- A language is a tool.
- Problems and Contexts
- We want to use the right tool for a given problem
- Communication

A LANGUAGE...

- Alphabet
- Constructs
- Syntax
- Semantics
- Type System (Static vs Dynamic)
- Translation Model (Compiled vs Interpreted)
- Execution model
- Runtime Mechanics
- Specification
- Implementation

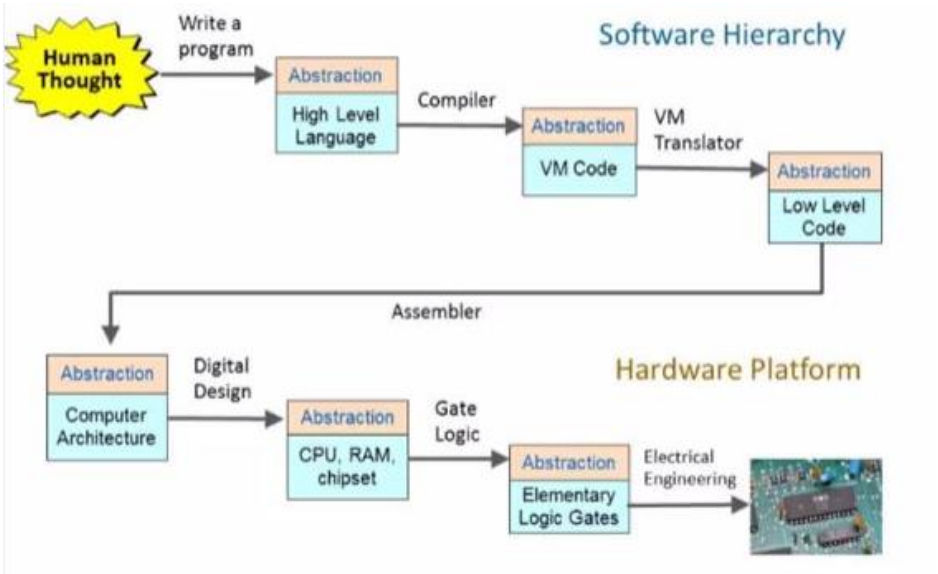
LANGUAGE TOOLS

- Editor
- Compiler
- REPL
- Virtual Machine
- Debugger
- Standard Library
- Dependency Management
- Build tool
- Workflow tools based on processes – Tests, Coverage, Syntax highlighting, Refactoring…

INSIDE THE HEAD OF A SOFTWARE ENGINEER

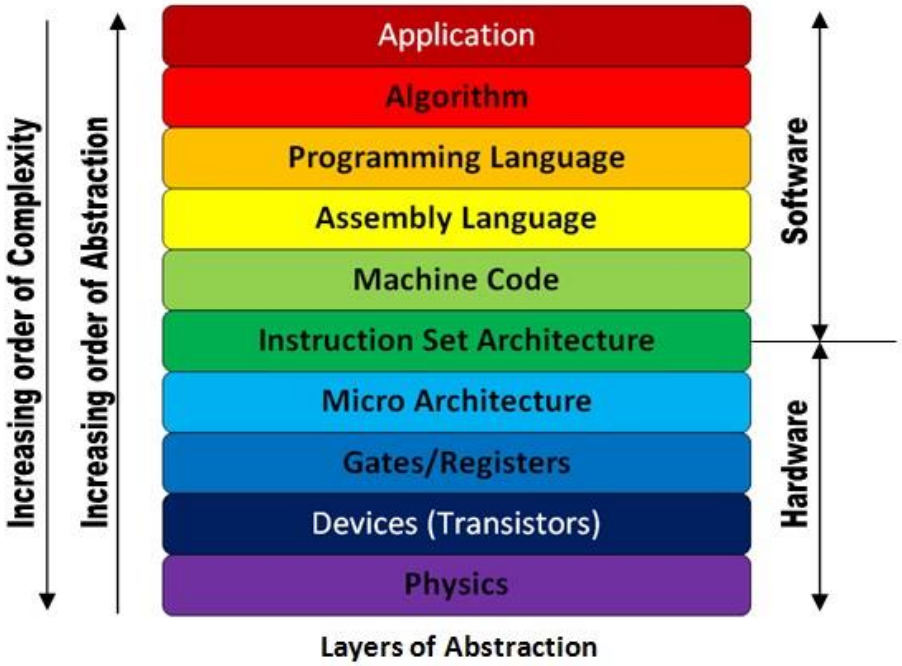
- Language Syntax
- Types
- Philosophy
- Paradigms
- Standards/Best Practices
- Principles
- Patterns
- Problem + Context
- Crafting solutions

THE PROCESS – FROM HUMAN THOUGHT TO SOMETHING MAGICAL



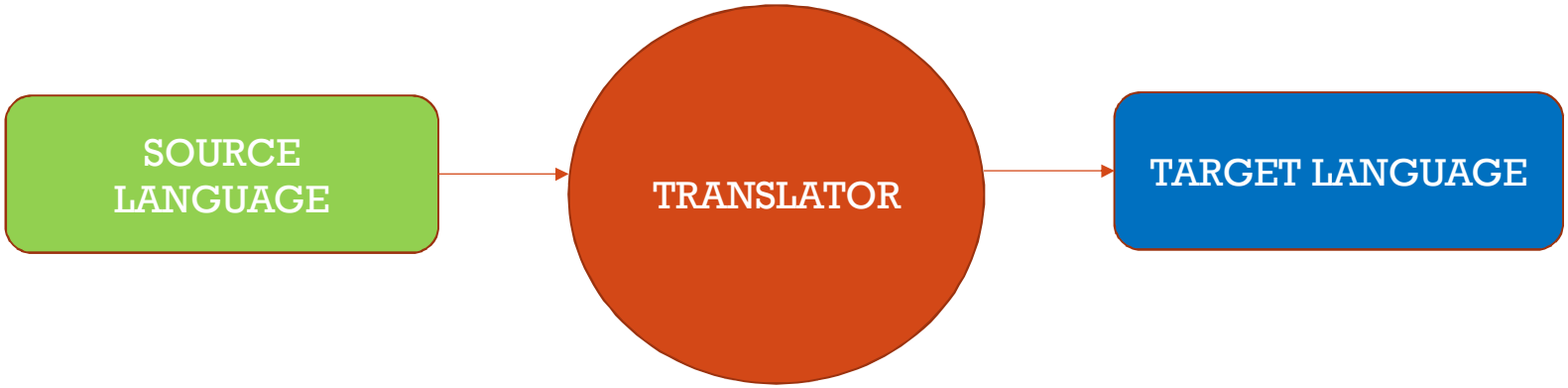
source : http://www.nand2tetris.org/

THE MACHINE STACK – LAYERS OF ABSTRACTION



source : http://theembeddedguy.com/2016/05/15/layers-of-abstraction/

LANGUAGE TRANSLATION



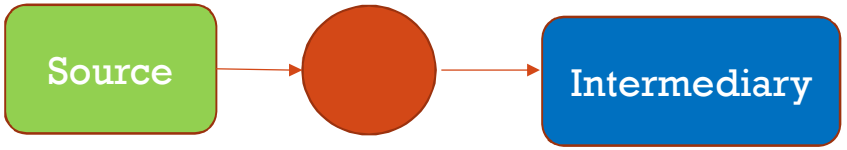
Based on the language, its philosophy, the translator could be called anything. Say, compiler for instance. A very important observation to make in the context of language translation is whether we can observe the Generation of an intermediary file or not. This also defines the execution/runtime mechanics.

MODERN LANGUAGES

- Roughly after 1985.
- Memory should be managed automatically. aka Garbage Collection
- Provision of higher level abstraction constructs.
- The concept of a runtime, intermediary language => Virtual Machine
- Developers Psychology - Discovery of patterns and anti patterns, How to write good/clean code?

VIRTUAL MACHINES

- Portability
- The notion of Intermediary Language
- Multiple Languages targeting the VM.
- Defines execution semantics



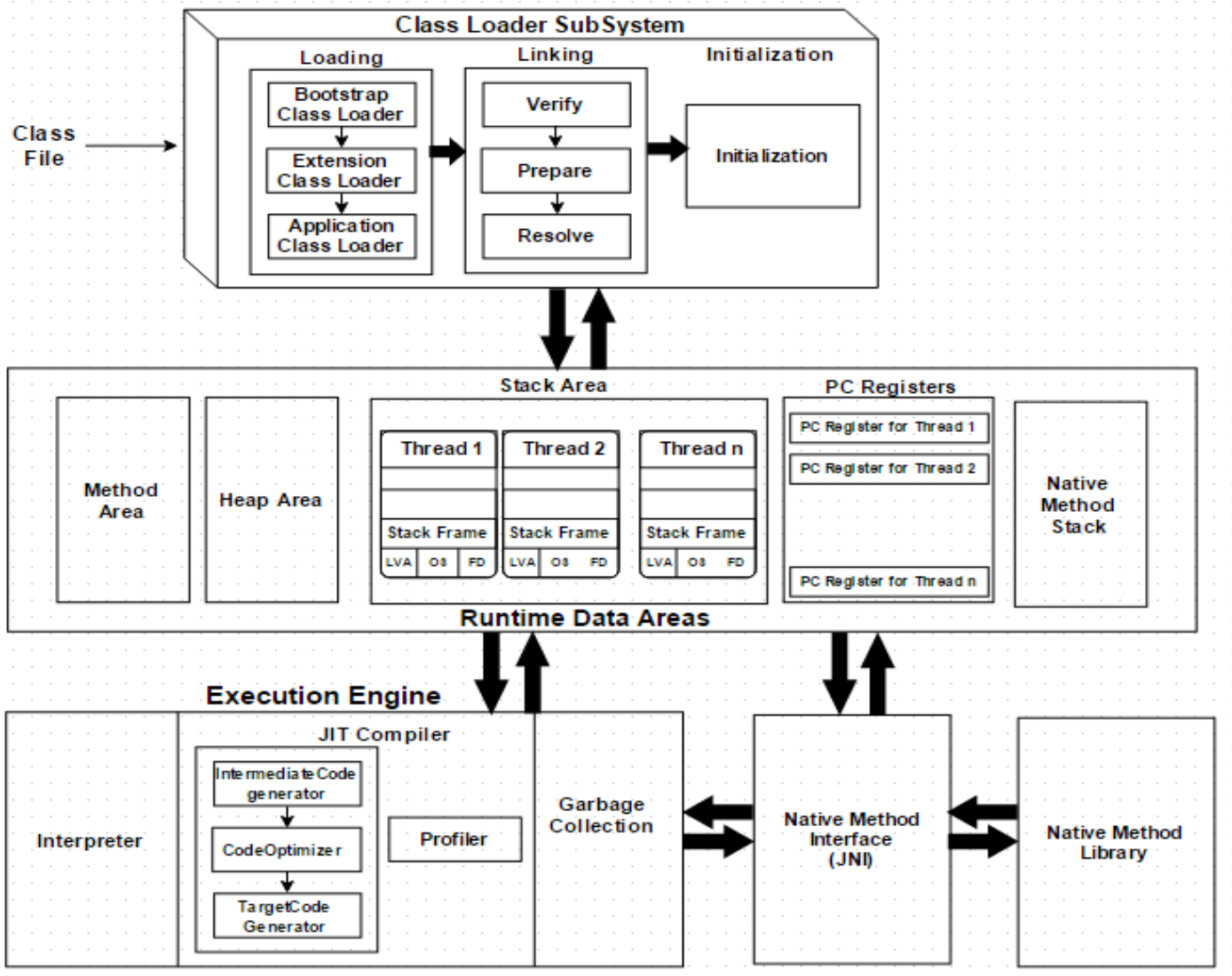
Translation process

RUNTIME ROLE

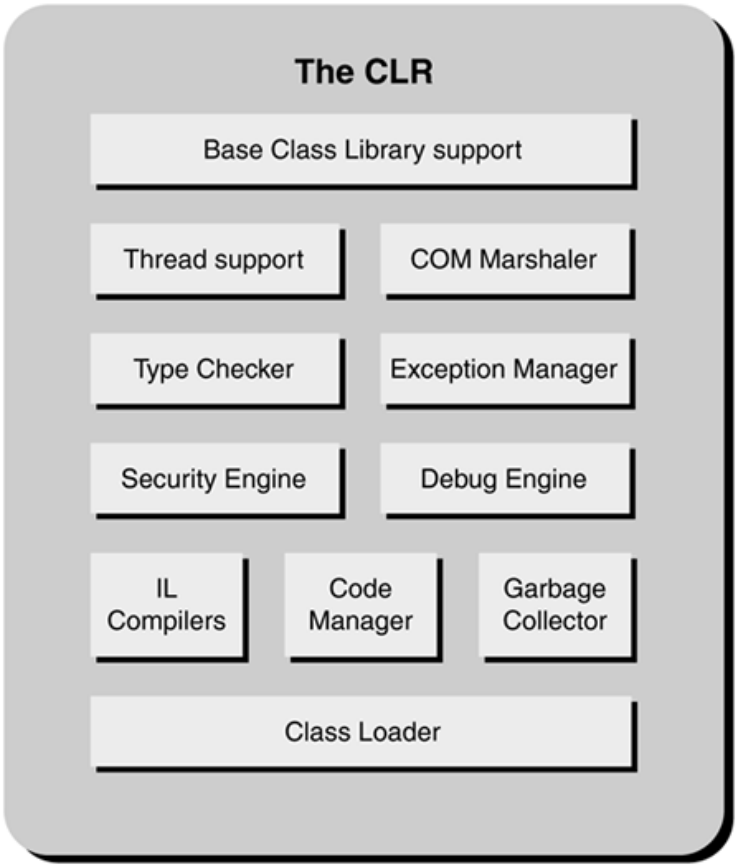
FAMILIAR VIRTUAL MACHINES

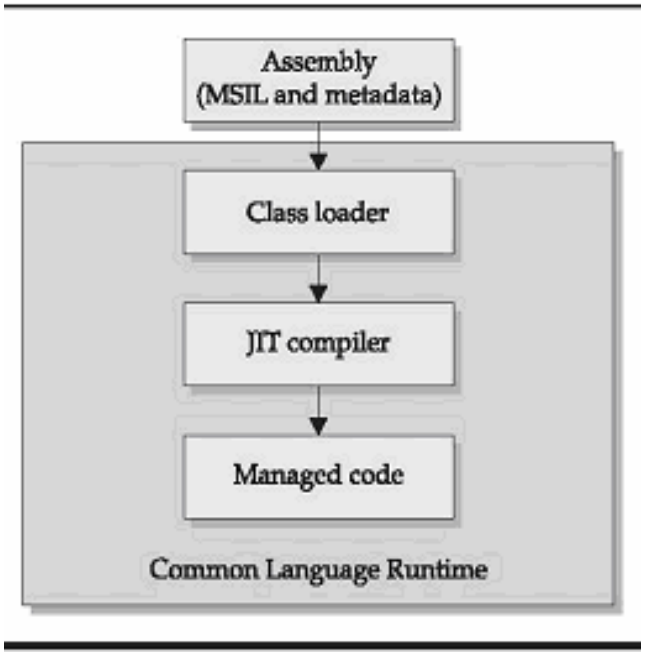
- Java World : Java Virtual Machine (JVM)
- .NET World: Common Language Runtime (CLR)

JVM

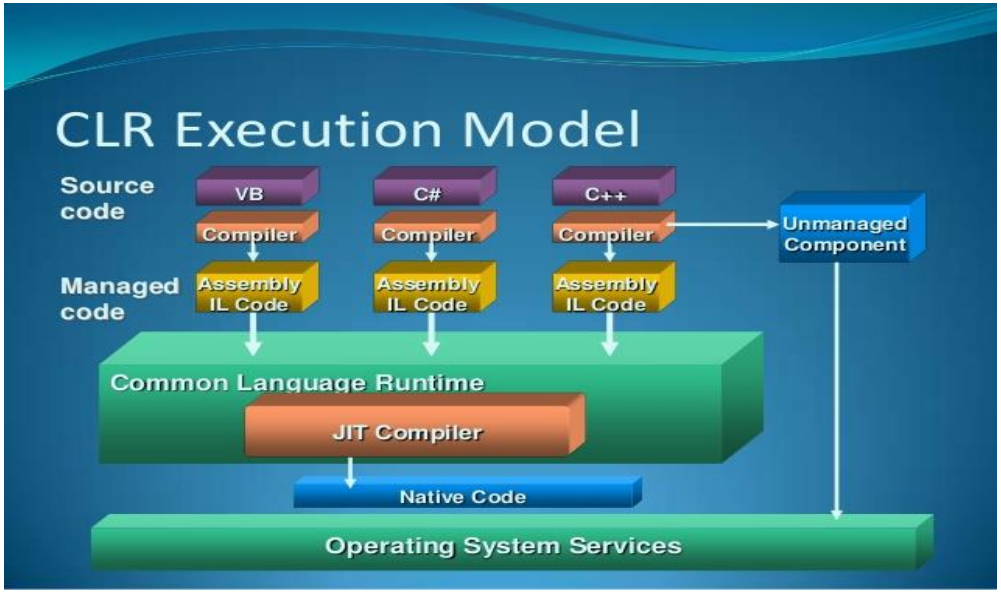

Class Loader SubSystem
Loading
Linking
Initialization
Class File
Bootstrap Class Loader
Extension Class Loader
Application Class Loader
Verify
Prepare
Resolve
Initialization
Stack Area
PC Registers
Method Area
Heap Area
Thread 1
Thread 2
Thread n
Stack Frame
LVA
OS
FD
PC Register for Thread 1
PC Register for Thread 2
PC Register for Thread n
Native Method Stack
Runtime Data Areas
Execution Engine
JIT Compiler
Intermediate Code generator
CodeOptimizer
TargetCode Generator
Profiler
Interpreter
Garbage Collection
Native Method Interface (JNI)
Native Method Library

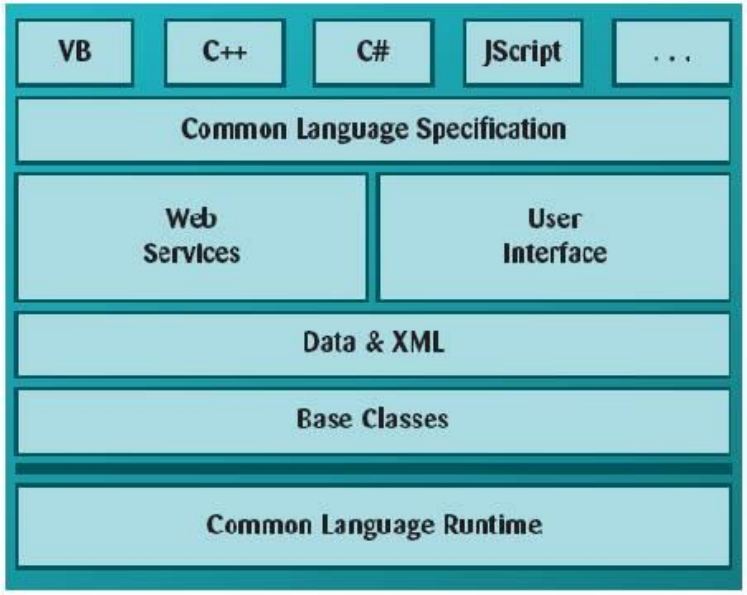
CLR


The CLR
Base Class Library support
Thread support
COM Marshaler
Type Checker
Exception Manager
Security Engine
Debug Engine
IL Compilers
Code Manager
Garbage Collector
Class Loader


Assembly (MSIL and metadata)
Class loader
JIT compiler
Managed code
Common Language Runtime

THE POWER OF AN INTERMEDIARY LANGUAGE & VM





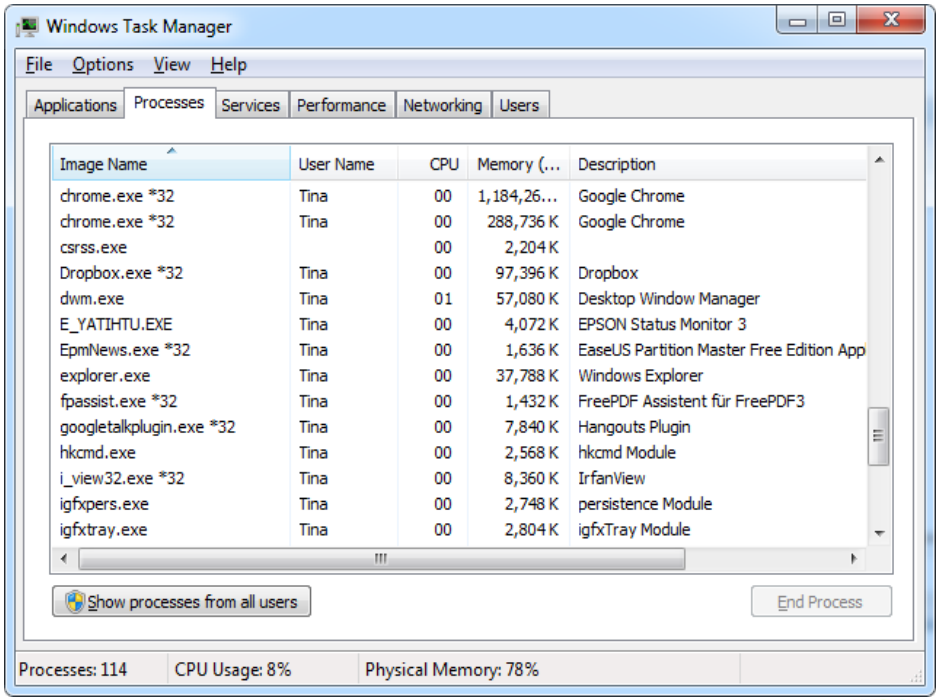
Multiple languages can now target the intermediary language. They can also all use the Services that the runtime has to offer. A high degree of reusability.

WHERE DOES THE OS FIT IN?

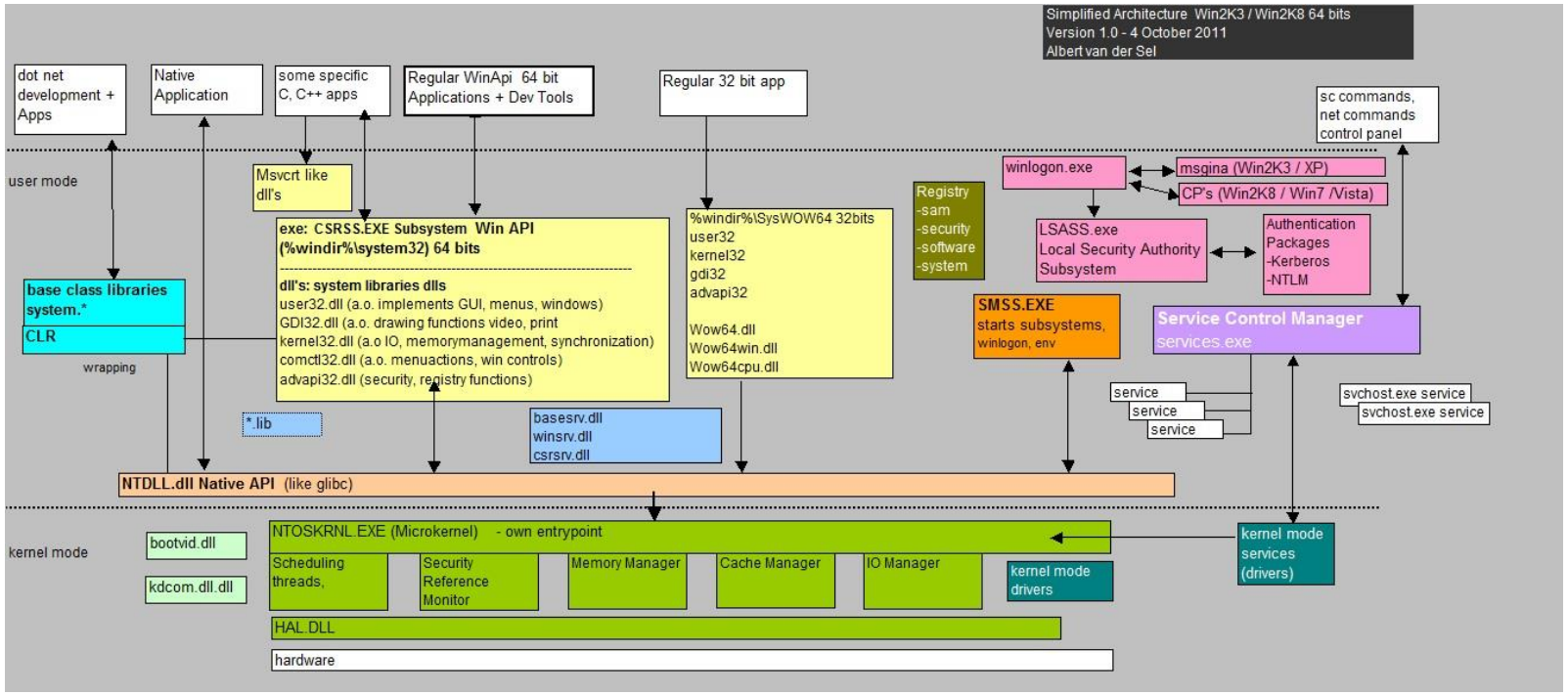
- VM/Runtime is all good.
- But…
- The fundamental way to run a program is via the OS.
- All hail the OS!!

OPERATING SYSTEM CONSTRUCTS

- Process
- Thread
- Concurrency
- Context Switching
- Schedulers
- Scheduling Algorithms
- The user is tricked. But that is good.
- Everyone loves Magic!



OPERATING SYSTEM OPERATION



HOW LANGUAGES ❤ EACH OTHER

- Languages influence one another.
- That means language designers(people) are constantly trying to improve the language
- They do this by exploring other languages/environments trying to find good ideas and bring them aboard.

JavaScript

Paradigm	Multi-paradigm: object-oriented (prototype-based), imperative, functional, event-driven[1]
Designed by	Brendan Eich
Developer	Netscape Communications Corporation, Mozilla Foundation, Ecma International
First appeared	December 4, 1995; 21 years ago[2]
Stable release	ECMAScript 2017[3] / June 2017; 1 month ago
Preview release	→←
Typing discipline	dynamic, duck
Filename extensions	`.js`
Website	Mozilla
Major implementations	
V8, JavaScriptCore, SpiderMonkey, Chakra	
Influenced by	
Lua, Scheme, Perl, Self, Java, C, Python, AWK, HyperTalk	
Influenced	
ActionScript, AtScript, CoffeeScript, Dart, JScript .NET, LiveScript, Objective-J, Opa, Perl 6, QML, TypeScript	

JavaScript at Wikibooks

THOUGHTS ON LANGUAGES

- I consider language as a tool, method to communicate our thoughts.
- Since it is a tool, it is a matter of preference.
- But, People are highly opinionated.
- People are programmed to think in one way, when they are in one environment.
- If we step out, we will find commonalities and find love for the language.
- Studying a lot of languages help us grow and think better.
- Languages are mostly multi-paradigm. That means we have a lot of thinking hats to put on and find the best way to solve a problem.
- We should continuously discover and enjoy
- Tooling is very important to build products quickly.
- Tooling has a "happy developer factor" associated with it.
- We should invest time in learning and learning is continuous.
- We should try to craft clean code.

THE ROAD TO ELIXIR...

CORE LEARNING INTERESTS

- Distributed Systems
- Event Oriented Architecture
- Message Oriented sytems
- Highly scalable systems
- Reactive Architectures

- "All the vocabulary is great! How should we learn these? What's the plan?" I said to myself.

THE BUS

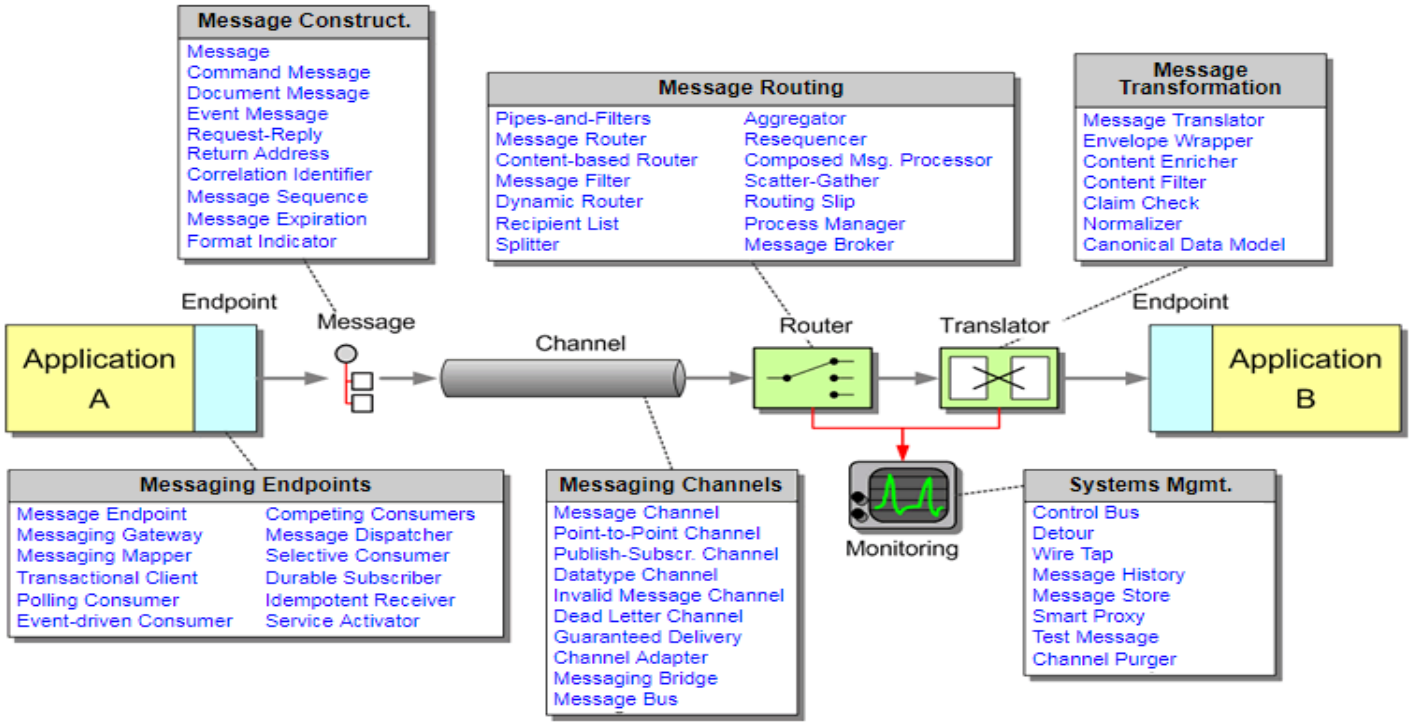
- Service Bus
- Enterprise Service Bus
- Message Bus
- I encountered a lot of terms.
- I worked on NServiceBus in my .NET days!

WHAT DOES IT MEAN TO BE A DISTRIBUTED SYSTEM?

- The 8 Fallacies of Distributed Computing
- https://en.wikipedia.org/wiki/Fallacies_of_distributed_computing
- Networks
- Interprocess communication
- Communication, Collaboration, Coordination.
- Application integration

INTEGRATION PATTERNS

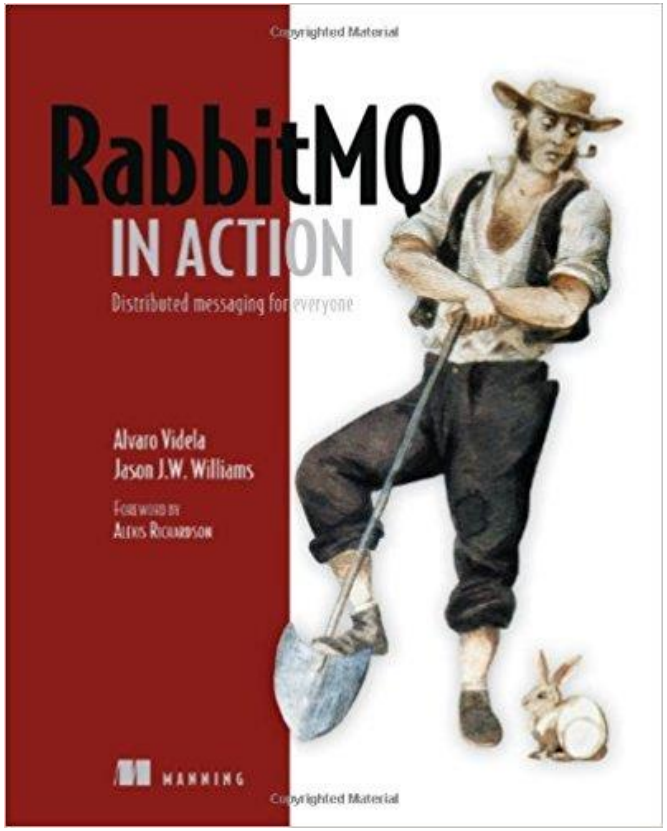
We have documented 65 messaging patterns, organized as follows:



http://www.enterpriseintegrationpatterns.com/patterns/messaging/

RABBITMQ ADVENTURES

- AMQP
- RabbitMQ history
- RabbitMQ Constructs
- RabbitMQ examples/patterns
- https://www.rabbitmq.com/getstarted.html



ERLANG

THE HISTORY OF ERLANG

- Ericsson
- Problem context
- The CS lab/team
- Solution Approach - RESEARCH -"Investigate the right programming language/environment for the given problem."
- 1982-85: "Experiments with programming of telecom using > 20 different languages. Conclusion: The language must be a very high level symbolic language in order to achieve productivity gains ! (Leaves us with: Lisp , Prolog , Parlog ...)"
- 1985-86: "Experiments with Lisp,Prolog, Parlog etc. Conclusion: The language must contain primitives for concurrency and error recovery, and the execution model must not have back-tracking. (Rules out Lisp and Prolog.) It must also have a granularity of concurrency such that one asyncronous telephony process is represented by one process in the language. (Rules out Parlog.)"
- Conclusion: "We must therefore **develop our own language with the desirable feature**s of Lisp, Prolog and Parlog, **but with concurrency and error recovery built into the language**."
- Source: https://www.erlang.org/course/history

LETS MEET THE PEOPLE

“Develop our own language with the desirable features of Lisp, Prolog and Parlog, **but with concurrency and error recovery built into the language.”**

IMPLEMENTATION

- 1987-93
- The notion of an "Abstract Machine" borrowed from Prolog.
- Joe's Abstract Machine
- Improve implementations through the years.
- Abstract Machine = Virtual Machine
- The Erlang Virtual Machine aka BEAM
- BEAM –

ERLANG OTP - OTHER AMAZING THOUGHTS

- Discover best practices, patterns
- Develop a set of tools.
- "OTP was originally meant to be – Open Telecom Platform"
- Coined during the age when "Open" was a common day-day use by Engineers.

- "OTP is set of Erlang libraries and design principles providing middle-ware to develop these systems. It includes its own distributed database, applications to interface towards other languages, debugging and release handling tools." – Erlang site

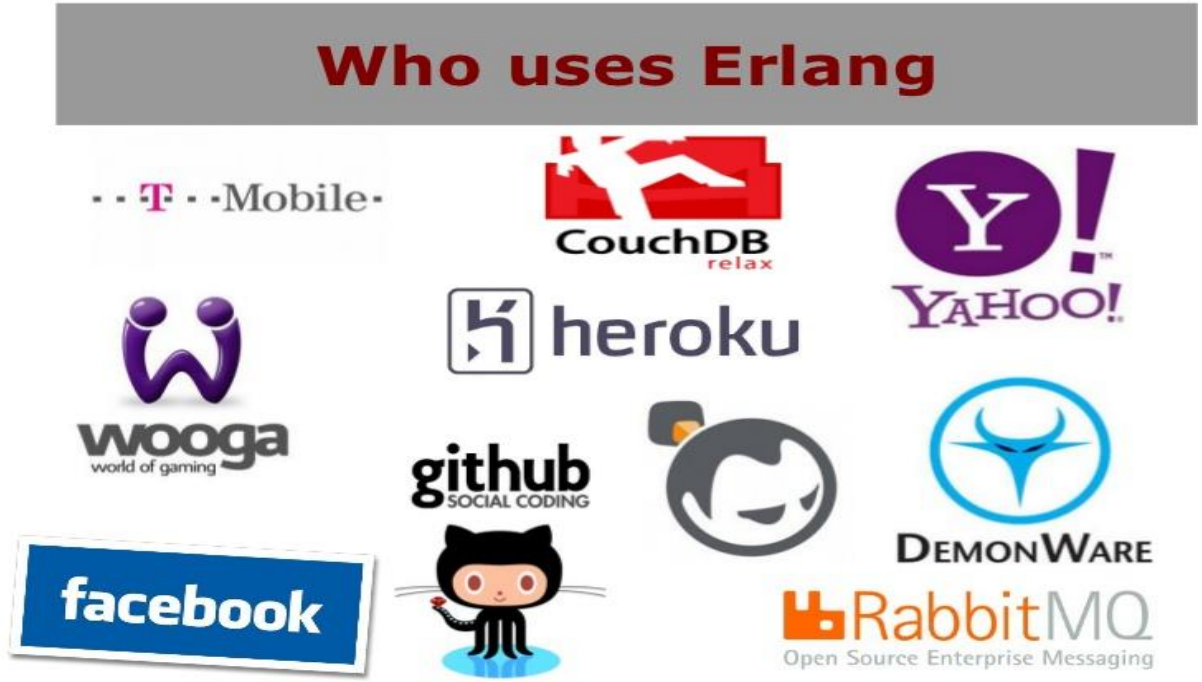
ERLANG

Erlang

Paradigm	multi-paradigm: concurrent, functional
Designed by	Joe Armstrong, Robert Virding, and Mike Williams
Developer	Ericsson
First appeared	1986; 31 years ago
Stable release	20.0[1] / 21 June 2017; 25 days ago
Typing discipline	dynamic, strong
License	Apache License 2.0 (since OTP 18.0) Erlang Public License 1.1 (earlier releases)
Filename extensions	.erl .hrl
Website	www.erlang.org
Major implementations	
Erlang	
Influenced by	
Prolog, Smalltalk, PLEX,[2] LISP	
Influenced	
F#, Clojure, Rust, Scala, Opa, Reia, Elixir, Dart, Akka	

Erlang Programming at Wikibooks

WHAT PROBLEMS DOES IT SOLVE?



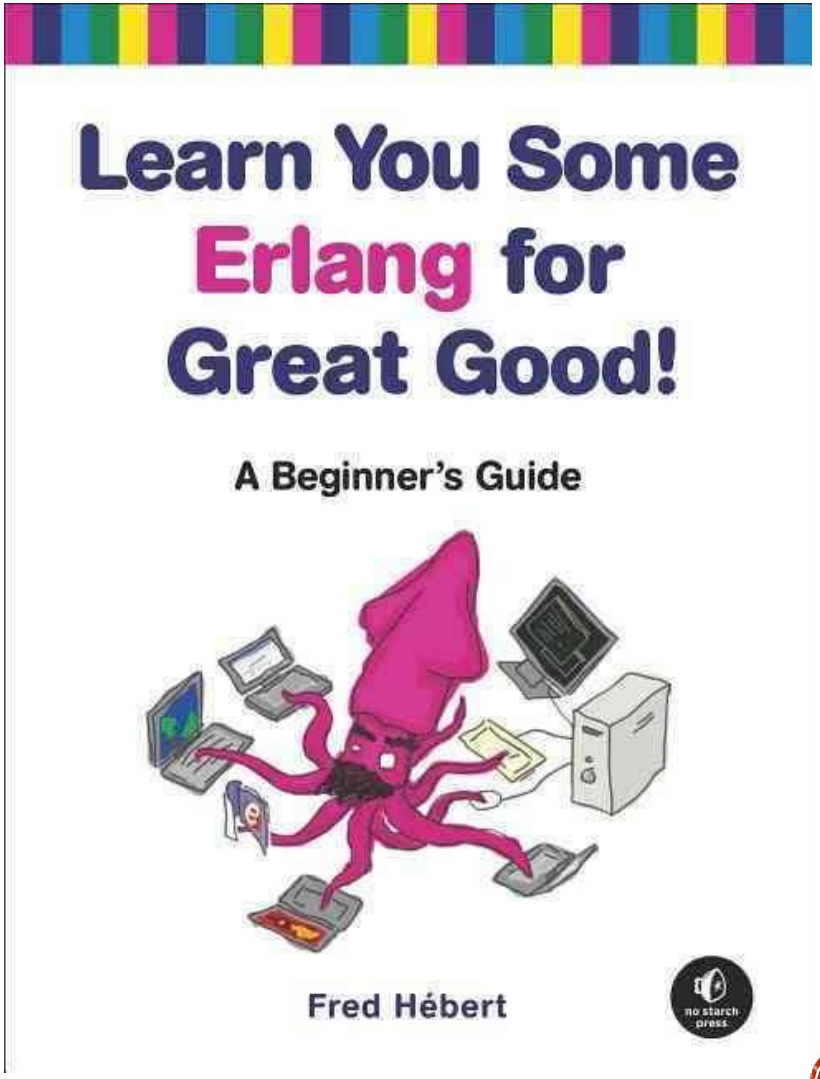
"By understanding who uses Erlang, we can understand the kinds of systems!"

SYSTEMS THAT ARE BASICALLY..

- "Massively scalable soft real-time systems with requirements on high availability"
- "Some of its uses are in telecoms, banking, e-commerce, computer telephony and instant messaging." – Erlang site
- Concurrent software, high level Concurrency

ACTOR MODEL

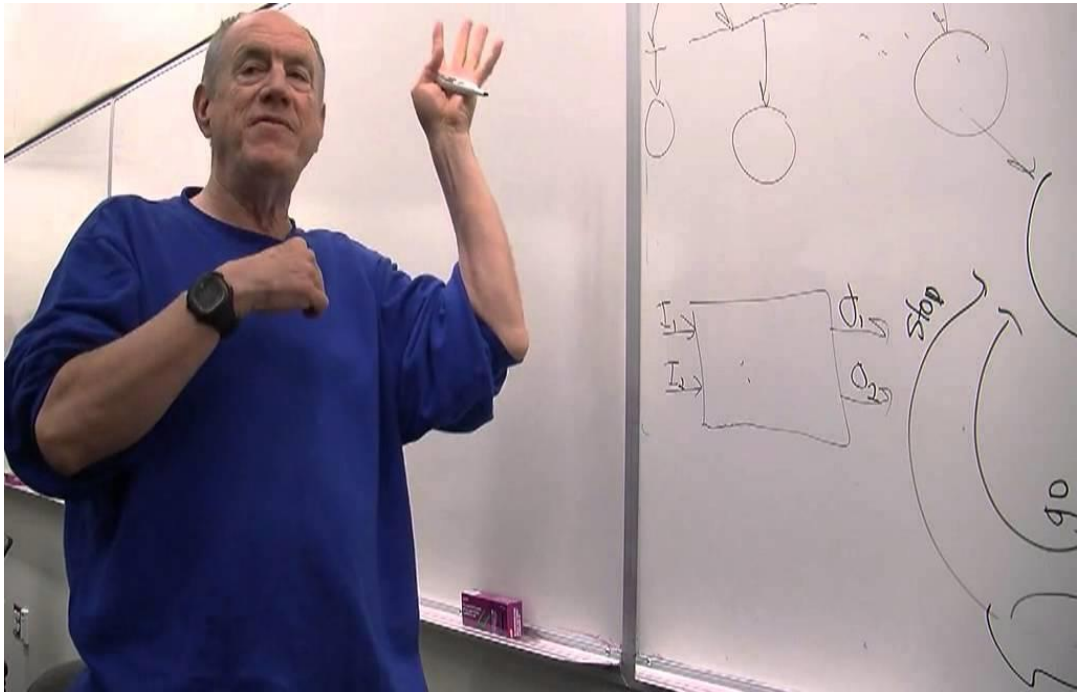
"Erlang uses the actor model, and each actor is a separate process in the virtual machine." – The book



ACTOR MODEL

- A model. Not a library. Not a technology.
- It is a way of thinking about "Computing"
- The **actor model** in computer science is a mathematical model of concurrent computation that treats "actors" as the universal primitives of concurrent computation. - Actor Model, Wikipedia
- The actor model originated in 1973.[1] It has been used both as a framework for a theoretical understanding of computation and as the theoretical basis for several practical implementations of concurrent systems. - Actor Model, Wikipedia

CARL HEWITT



"Carl Hewitt explaining the Actor model" - https://www.youtube.com/watch?v=7erJ1DV_Tlo&t=761s

CARL HEWITT SPEAKS

- According to Carl Hewitt, unlike previous models of computation, the actor model was inspired by physics, including general relativity and quantum mechanics. It was also influenced by the programming languages Lisp, Simula and early versions of Smalltalk, as well as capability-based systems and packet switching.
- Its development was "motivated by the prospect of highly parallel computing machines consisting of dozens, hundreds, or even thousands of independent microprocessors, each with its own local memory and communications processor, communicating via a high-performance communications network."[2] Since that time, the advent of massive concurrency through multi-core and manycore computer architectures has revived interest in the actor model.

ACTOR MODEL

- **The actor model adopts the philosophy that everything is an actor.**
- This is similar to the everything is an object philosophy used by some object-oriented programming languages.
- **An actor is a computational entity that, in response to a message it receives, can concurrently:**
- send a finite number of messages to other actors;
- create a finite number of new actors;
- designate the behavior to be used for the next message it receives.
- There is no assumed sequence to the above actions and they could be carried out in parallel.
- Decoupling the sender from communications sent was a fundamental advance of the Actor model enabling asynchronous communication and control structures as patterns of passing messages.[8]
- Recipients of messages are identified by address, sometimes called "mailing address". Thus an actor can only communicate with actors whose addresses it has. It can obtain those from a message it receives, or if the address is for an actor it has itself created.
- The actor model is characterized by inherent concurrency of computation within and among actors, dynamic creation of actors, inclusion of actor addresses in messages, and interaction only through direct asynchronous message passing with no restriction on message arrival order.

ERLANG REVISITED

- TWO PARTS TO ERLANG
- Sequential Programming
- Concurrent Programming

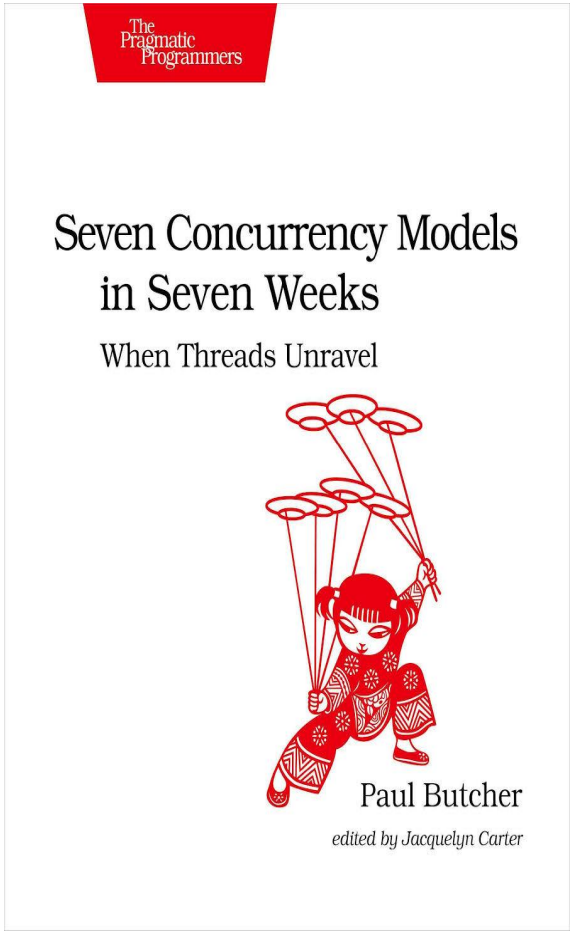
CONCURRENCY ORIENTED PROGRAMMING

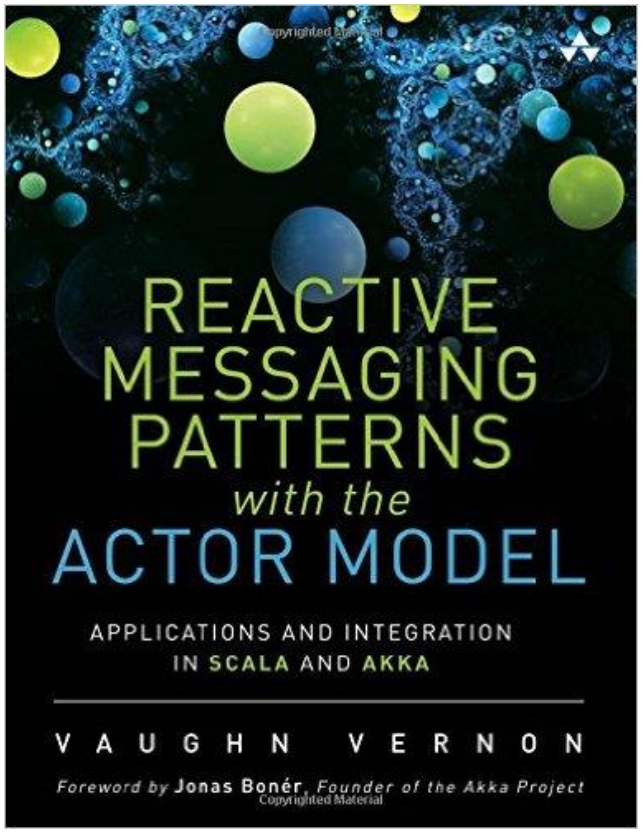
- Concurrency is key in the language.
- The language needs to provide '**constructs**' to support concurrency!
- Erlang constructs are called '**processes**'.
- **Processes communicate with each other by passing messages to one another.**
- To achieve concurrency, "**schedulers**" are in place.

DOES THAT MEAN ERLANG WAS IMPLEMENTING CARL HEWITT'S ACTOR MODEL?

- NO.
- Sweden vs US
- No Web at the time.
- It just maps neatly! That is all.

- "We did not know about the Actor model then." – Joe Armstrong


The Pragmatic Programmers
Seven Concurrency Models in Seven Weeks
When Threads Unravel
Paul Butcher
edited by Jacquelyn Carter


REACTIVE MESSAGING PATTERNS with the ACTOR MODEL
APPLICATIONS AND INTEGRATION IN SCALA AND AKKA
VAUGHN VERNON
Foreword by Jonas Bonér, Founder of the Akka Project
Copyrighted Material

ALAN KAY

"I made up the term 'object-oriented', and I can tell you I didn't have C++ in mind"
-- Alan Kay, OOPSLA '97

Alan Kay

Kay at the 2008 40th anniversary of The Mother of All Demos

Born	Alan Curtis Kay May 17, 1940 (age 77) Springfield, Massachusetts
Citizenship	United States
Fields	Computer science
Institutions	Xerox PARC Stanford University Atari Apple Inc. ATG Walt Disney Imagineering UCLA Kyoto University MIT Viewpoints Research Institute Hewlett-Packard Labs
Alma mater	University of Colorado at Boulder, University of Utah
Doctoral advisor	David C. Evans Robert S. Barton
Known for	Dynabook object-oriented programming Smalltalk graphical user interface windows
Notable awards	ACM Turing Award (2003) Kyoto Prize Charles Stark Draper Prize
Spouse	Bonnie MacBird

ALAN KAY ON OBJECT ORIENTATION

"The big idea is messaging" – Alan Kay

AKKA

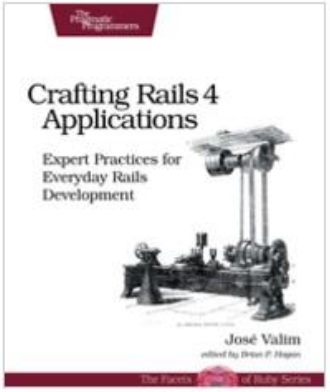
Akka

Original author(s)	Jonas Bonér
Developer(s)	Lightbend
Initial release	July 2009
Stable release	2.5.1[1] / May 2, 2017[2]
Repository	github.com/akka/akka
Development status	Active
Written in	Scala
Operating system	Cross-platform
Platform	Java Virtual Machine
License	Apache License 2.0
Website	akka.io

- "**Akka** is a free and open-source toolkit and runtime simplifying the construction of concurrent and distributed applications on the JVM. Akka supports multiple programming models for concurrency, but it emphasizes actor-based concurrency, with inspiration drawn from Erlang." – Akka, Wikipedia
- http://akka.io/, http://getakka.net/

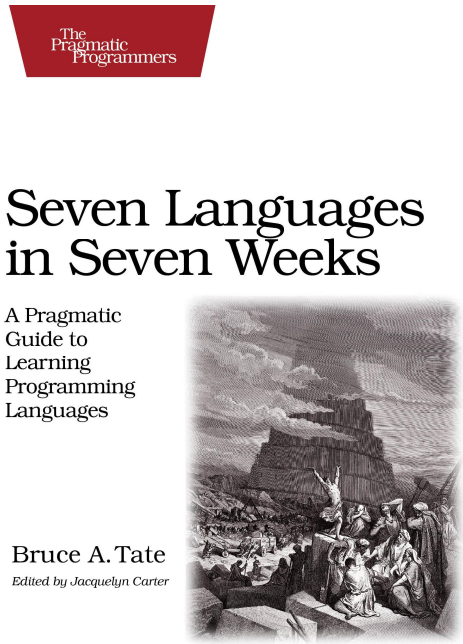
JOSE VALIM

SAY HI TO JOSE VALIM



RUBY ON RAILS CORE CONTRIBUTOR

I NEED "HIGH PERFORMANT SOFTWARE" - JOSE



IN LOVE WITH ERLANG

- Jose loves the technology!
- He loves the concepts!
- He chooses to adopt the "Erlang VM" to his toolset.
- But after sometime…
- Starts thinking hard!
- I need better tooling.
- I need a better syntax.

ELIXIR

Elixir

elixir

Paradigm	multi-paradigm: functional, concurrent, distributed, process-oriented
First appeared	2011; 6 years ago
Stable release	1.4.5 / 22 June 2017; 24 days ago[1]
Typing discipline	dynamic, strong
Platform	Erlang
License	Apache License 2.0[2]
Filename extensions	.ex, .exs
Website	elixir-lang.org
Influenced by	
Erlang, Ruby, Clojure	
Influenced	
LFE	

ELIXIR

- Built on top of the Erlang VM
- Modern syntax
- Aggregated ideas from various programming languages.
- Interoperable with Erlang code.
- All benefits from OTP free.
- Builds on top of Erlang constructs and provides cleaner abstractions.
- Amazing tooling.
- Sophisticated ecosystem.
- Awesome community.

ELIXIR DISTRIBUTION

Stable (download)

- Elixir - standard library
- EEx - templating library
- ExUnit - unit test library
- IEx - interactive shell
- Logger - built-in Logger
- Mix - build tool

- Tools – elixir, elixirc, iex
- Standard Library
- Mix – The comprehensive build system : Best practices, code organization, code generation
- Hex – Package Management
- ExUnit – Unit testing
- ExDocs – Documentation

ELIXIR IN ACTION...

AGENDA

- Understand Elixir dependencies
- Distribution - Observe files on the disk
- Iex
- :observer

`String.length/1` will take longer as the input grows.

CODE CONSTRUCTS

- Module
- Function
- The |> operator
- Data – Atoms, Tuples, Lists, Maps, Binaries, Strings, Char lists, Keyword lists
- Pattern Matching
- '=' does not mean assignment. It means binding.
- Reading docs

MODULES

- Modules are code organization units
- We can have nested modules
- Nested Modules have

FUNCTIONS

- Belong to a Module
- You can have private and public functions
- def, defp
- String.length/1

THE POWER OF ELIXIR

- Utilizing the Erlang VM
- Agent, GenServer, Supervision
- Metaprogramming (Using Abstract Syntax Tree as code)

CODE EXAMPLE

- mix –help
- mix new

ECOSYSTEM

RESOURCES TO LEARN ELIXIR

- https://github.com/sameeri/Hello-Elixir