

HELLO ELIXIR!

Sameeri Marryboyina





Premier Sponsors



Gold Sponsors



Silver Sponsors



Lunch Sponsors



Resource Sponsors



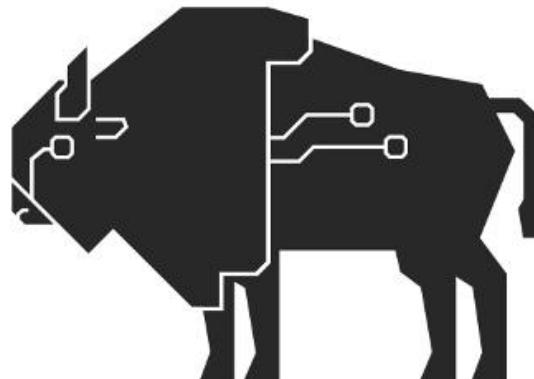
Prize Sponsors



THANK YOU TECHLAHOMA!



OKC FP



TECHLAHOMA

OKC Elixir

Find Techlahoma on all things social.



TALK AGENDA

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



ABOUT ME



HELLO, SAMEERI.

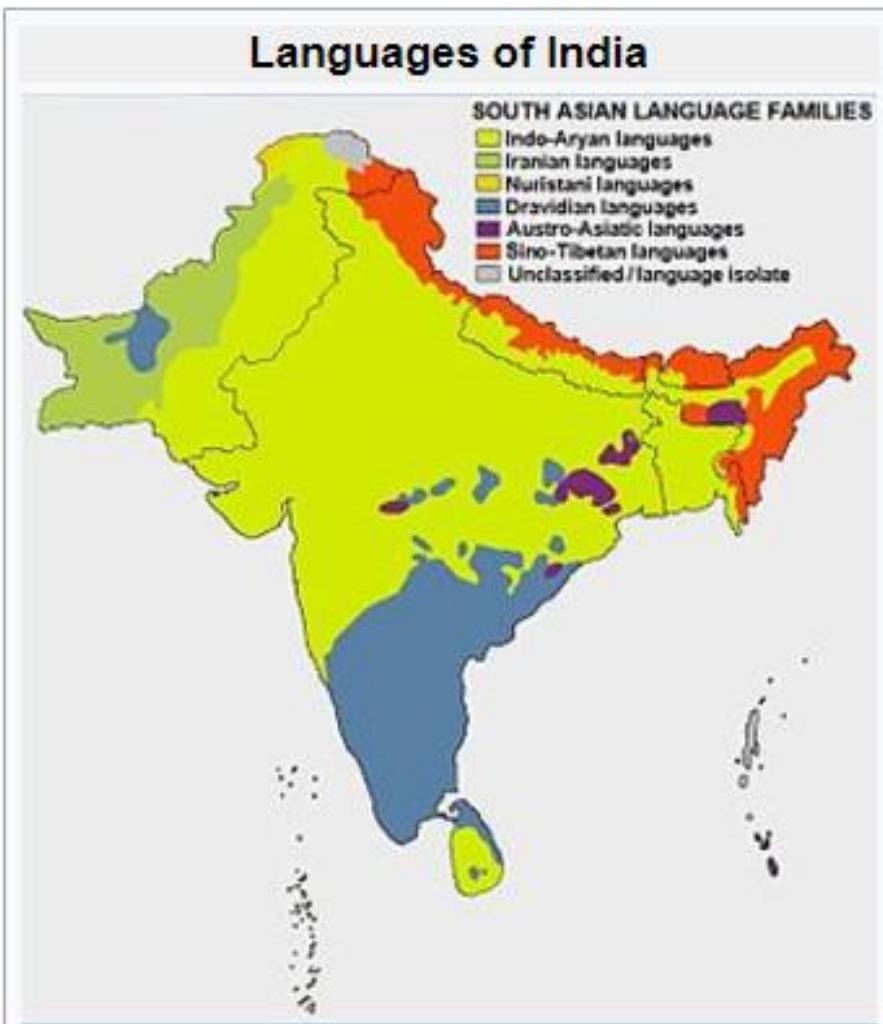
- Sameeri Pavan Kumar Marryboyina
- Hyderabad, India
- I can communicate in Telugu, Hindi, English, German(very little)
- MS in Computer Science, The University of Oklahoma
- Senior Software Engineer, Telogical Systems LLC
- Full Stack JS - React, Redux and its Ecosystem, Node, Express
- Norman, OK



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	<ul style="list-style-type: none"> • English – around 125 million speakers^[5] • French – approximately 75,000 speakers

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



TARGET AUDIENCE



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...

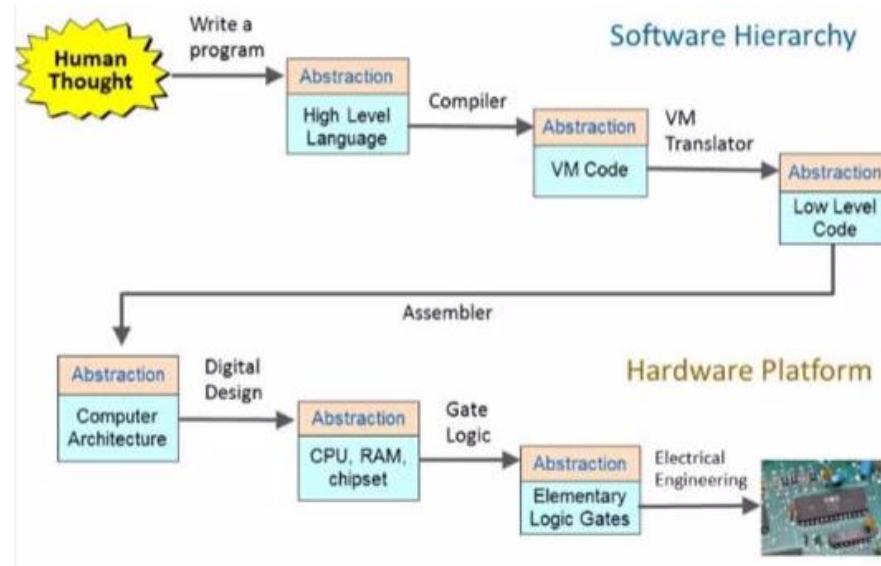


INSIDE THE HEAD OF A SOFTWARE ENGINEER

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



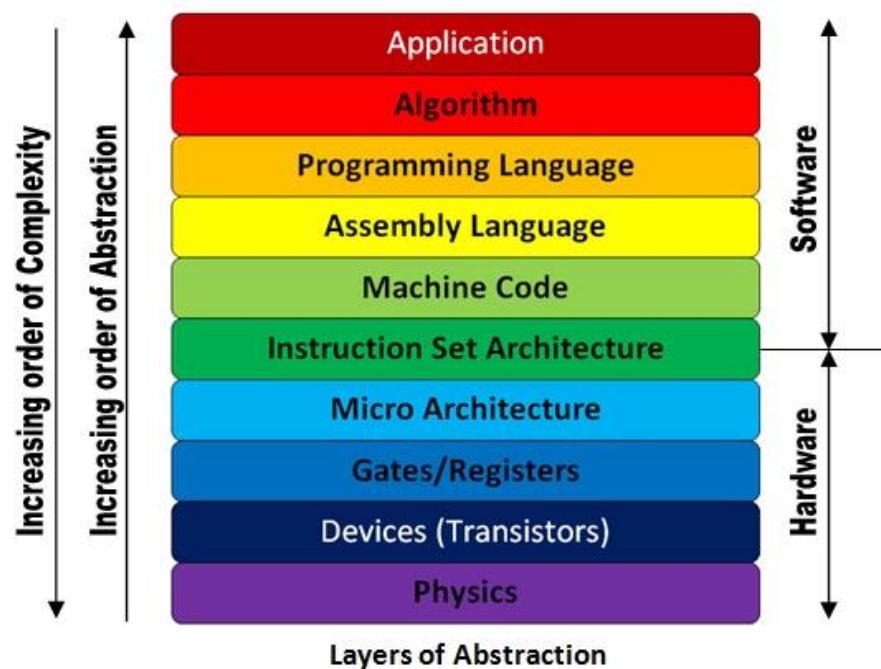
THE PROCESS – FROM HUMAN THOUGHT TO SOMETHING MAGICAL



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



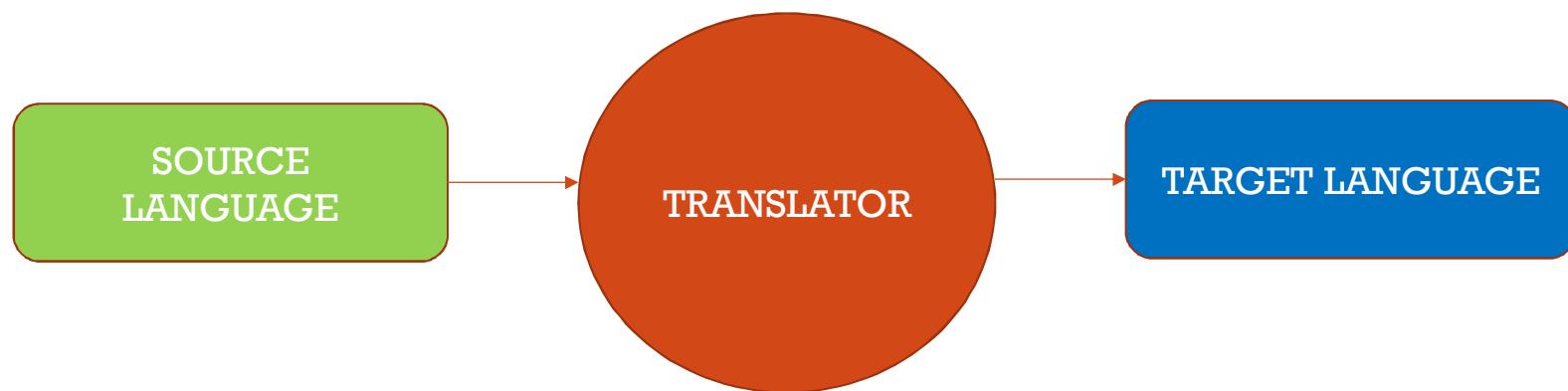
THE MACHINE STACK – LAYERS OF ABSTRACTION



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.



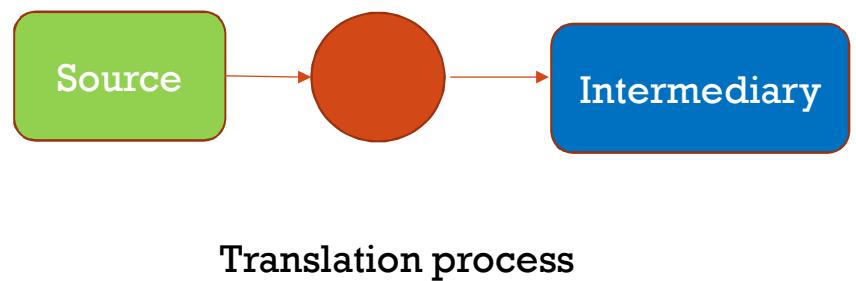
MODERN LANGUAGES

- 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



VIRTUAL MACHINE

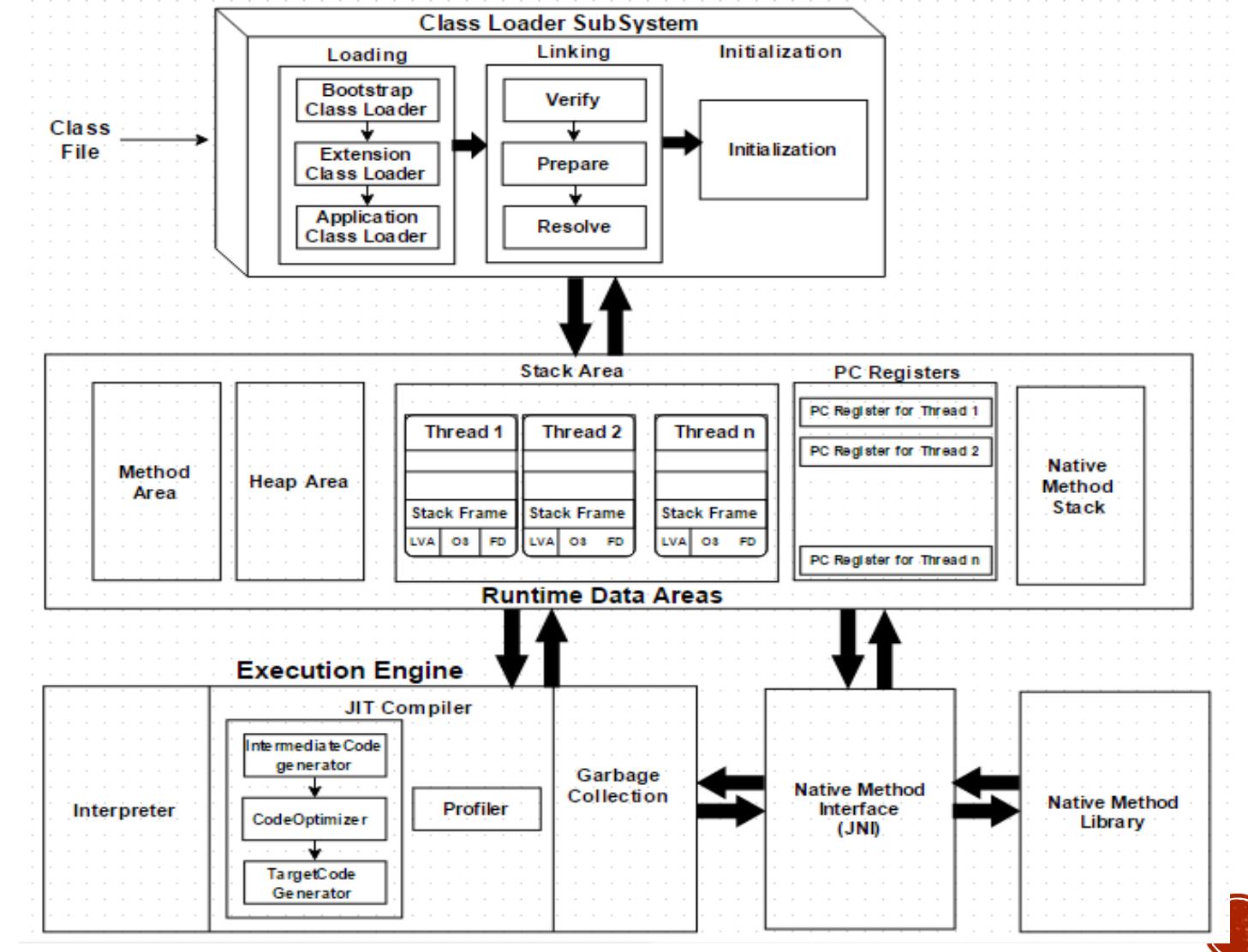


FAMILIAR VIRTUAL MACHINES

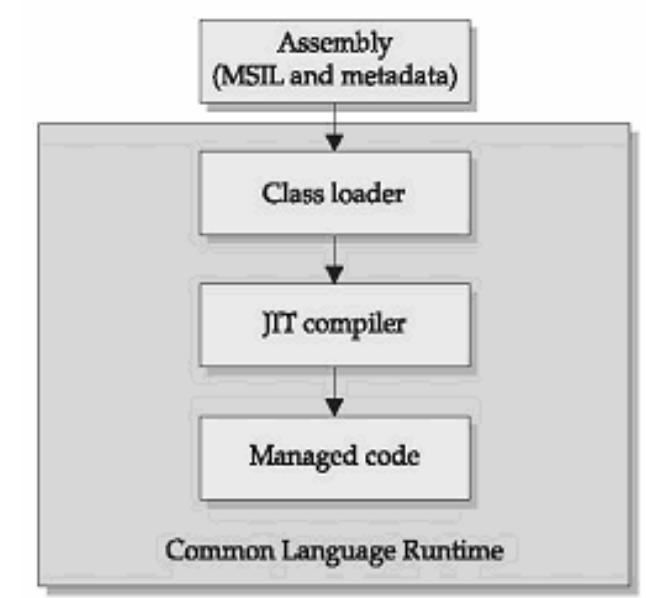
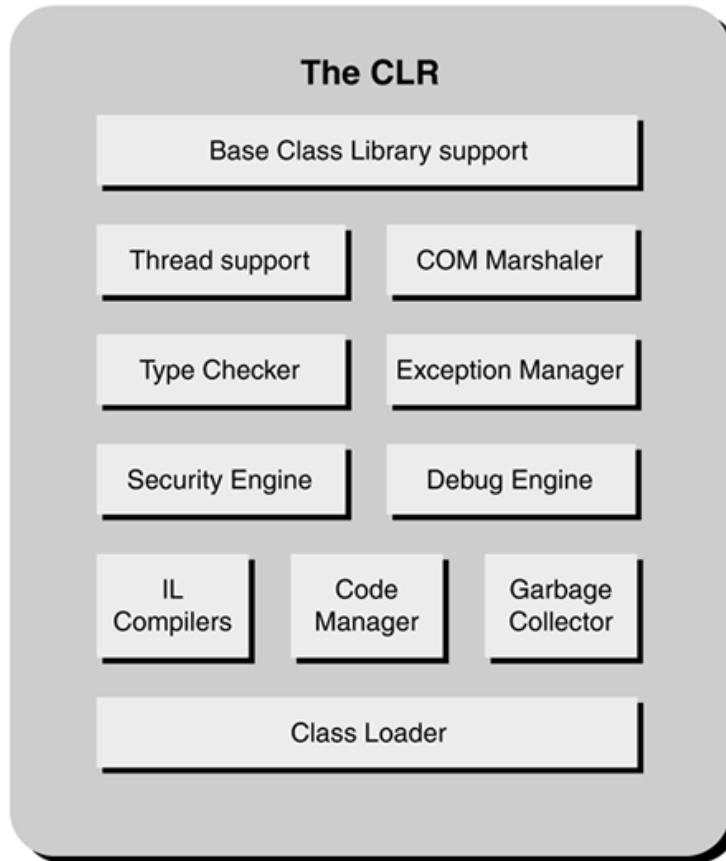
- Java World : Java Virtual Machine (JVM)
- .NET World: Common Language Runtime (CLR)



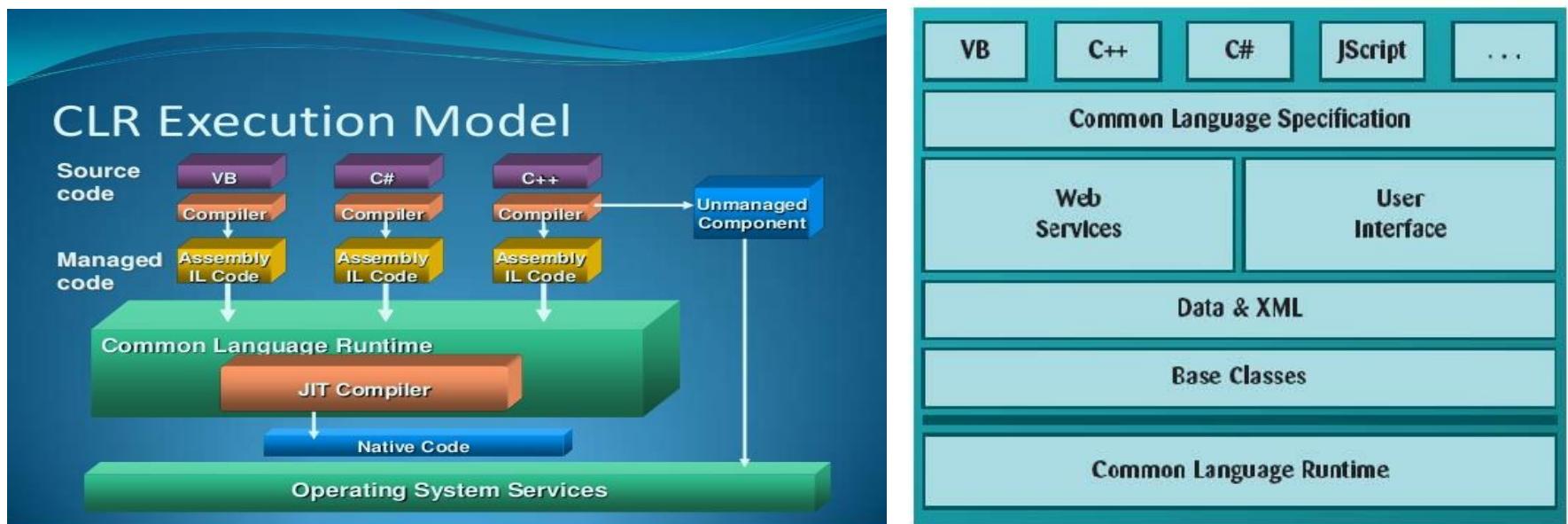
JVM



CLR



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.



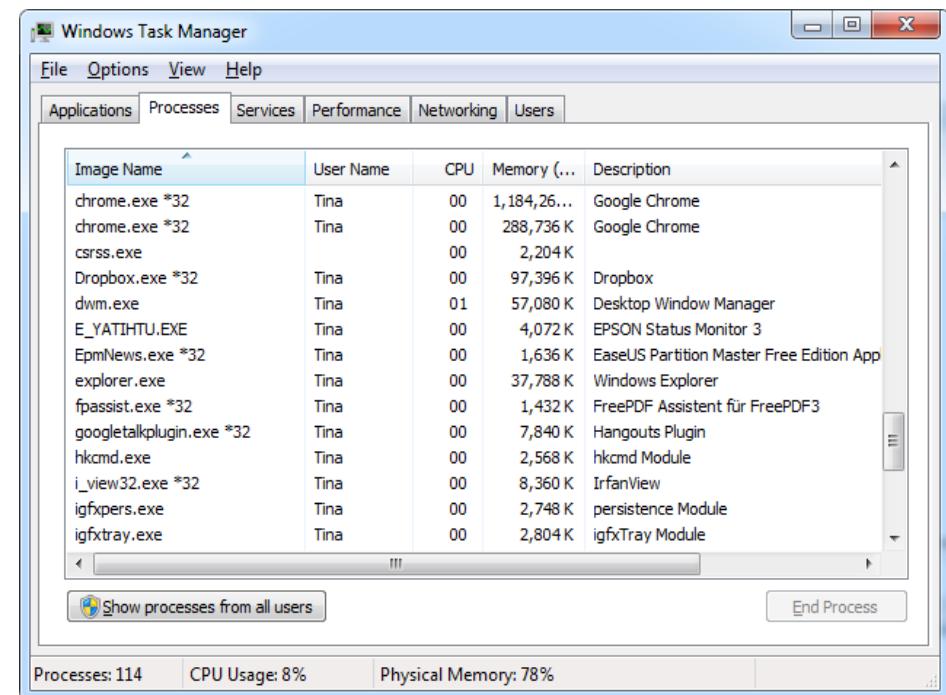
WHERE DOES THE OS FIT IN?

- 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!

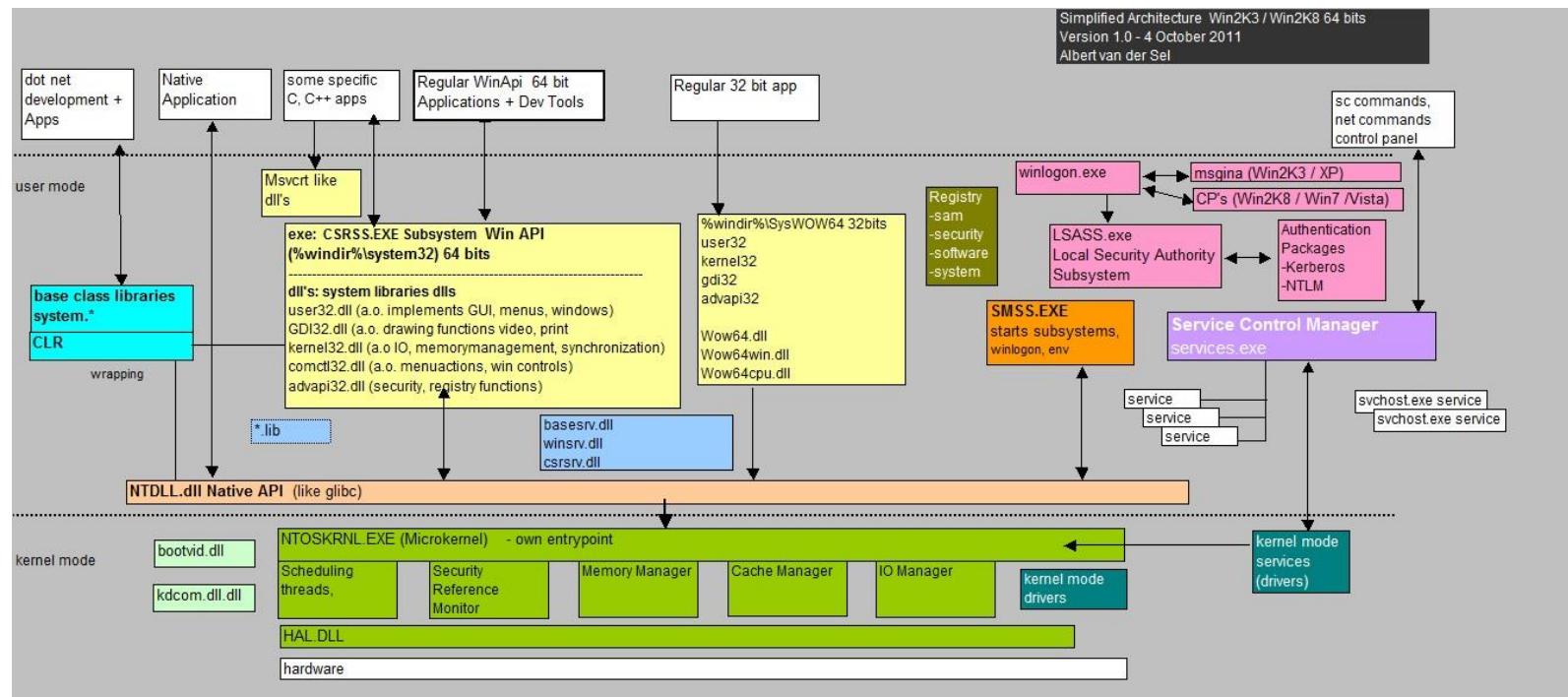


A screenshot of the Windows Task Manager window titled "Windows Task Manager". The "Processes" tab is selected. The table lists various processes with their names, users, CPU usage, memory usage, and descriptions. The processes listed include chrome.exe, csrss.exe, Dropbox.exe, dwm.exe, E_YATHTU.EXE, EpmNews.exe, explorer.exe, fpassist.exe, googletalkplugin.exe, hkcmd.exe, i_view32.exe, igfxpers.exe, and igfxtray.exe. The "User Name" column shows "Tina" for most processes, except for the system ones like csrss.exe and dwm.exe. The "CPU" column shows values like 00, 01, and 00. The "Memory" column shows values like 1,184,26..., 288,736 K, and 2,204 K. The "Description" column provides the application names. At the bottom of the window, there are buttons for "Show processes from all users", "End Process", and performance metrics: "Processes: 114", "CPU Usage: 8%", and "Physical Memory: 78%".

Image Name	User Name	CPU	Memory (...)	Description
chrome.exe *32	Tina	00	1,184,26...	Google Chrome
chrome.exe *32	Tina	00	288,736 K	Google Chrome
csrss.exe		00	2,204 K	
Dropbox.exe *32	Tina	00	97,396 K	Dropbox
dwm.exe	Tina	01	57,080 K	Desktop Window Manager
E_YATHTU.EXE	Tina	00	4,072 K	EPSON Status Monitor 3
EpmNews.exe *32	Tina	00	1,636 K	EaseUS Partition Master Free Edition App
explorer.exe	Tina	00	37,788 K	Windows Explorer
fpassist.exe *32	Tina	00	1,432 K	FreePDF Assistant für FreePDF3
googletalkplugin.exe *32	Tina	00	7,840 K	Hangouts Plugin
hkcmd.exe	Tina	00	2,568 K	hkcmd Module
i_view32.exe *32	Tina	00	8,360 K	IrfanView
igfxpers.exe	Tina	00	2,748 K	persistence Module
igfxtray.exe	Tina	00	2,804 K	igfxTray Module



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 systems
- 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!



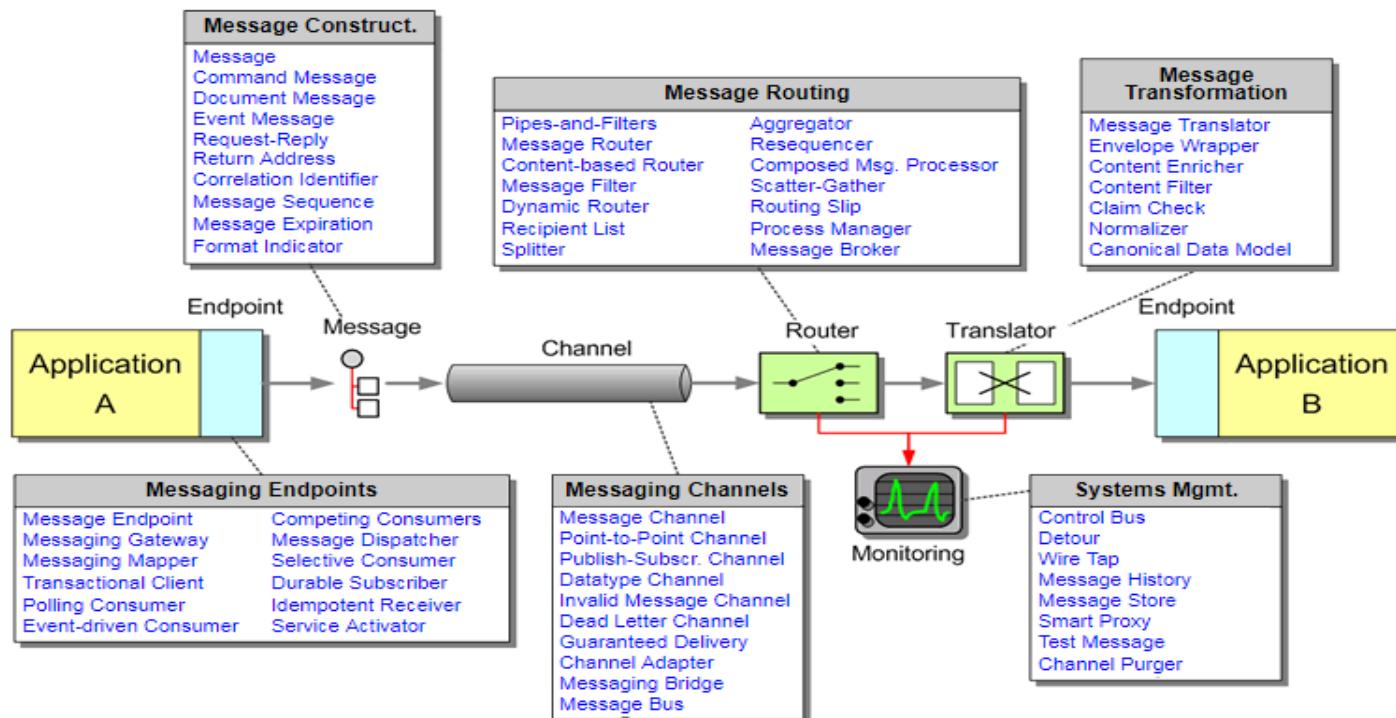
WHAT DOES IT MEAN TO BE A DISTRIBUTED SYSTEM?

- 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:

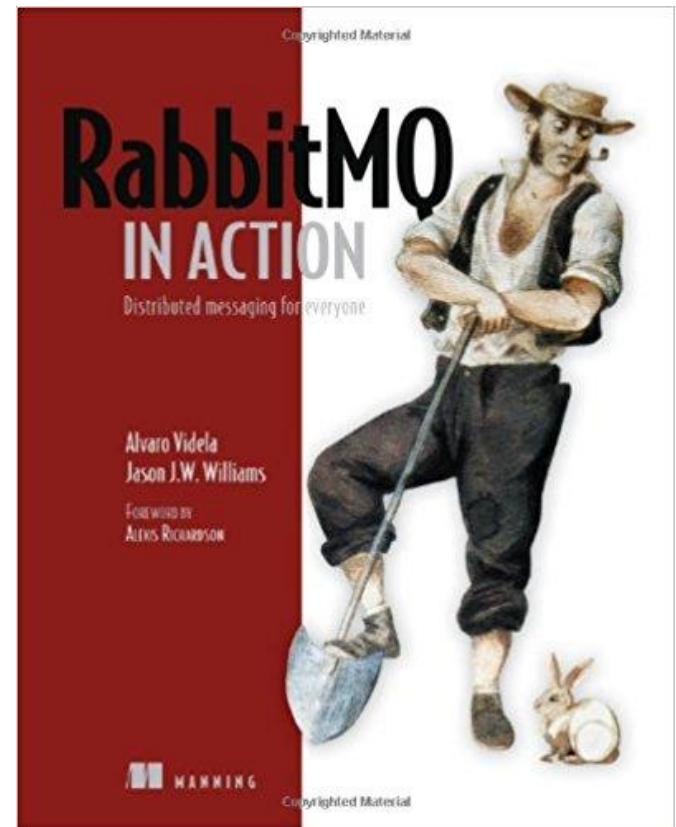


<http://www.enterpriseintegrationpatterns.com/patterns/messaging/>



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 asynchronous telephony process is represented by one process in the language. (Rules out Parlog.)”
- Conclusion: “We must therefore **develop our own language with the desirable features** of Lisp, Prolog and Parlog, **but with concurrency and error recovery built into the language.**”
- Source: <https://www.erlang.org/course/history>



LET'S MEET THE PEOPLE



Robert Virding

Creator of Erlang

Mike Williams

Creator of Erlang

Joe Armstrong

Creator of Erlang

“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?

Who uses Erlang

...T...Mobile



"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

Learn You Some Erlang for Great Good!

A Beginner’s Guide



Fred Hébert

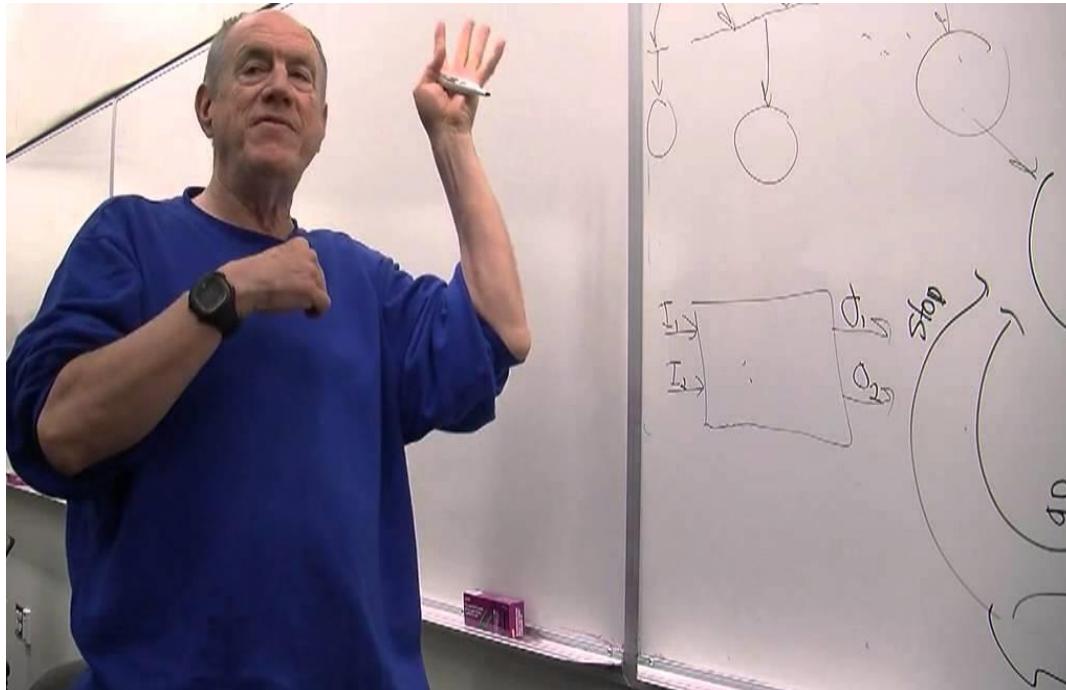


ACTOR MODEL

- 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



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

 **Alan Kay On Messaging**

From: Alan Kay <alan_kay@wdi.disney.com>
Date: 1998-10-10 07:39:40 +0200
To: squeak@cs.uiuc.edu
Subject: Re: prototypes vs classes was: Re: Sun's HotSpot

Folks --

Just a gentle reminder that I took some pains at the last OOPSLA to try to remind everyone that Smalltalk is not only NOT its syntax or the class library, it is not even about classes. I'm sorry that I long ago coined the term "objects" for this topic because it gets many people to focus on the lesser idea.

The big idea is "messaging" - that is what the kernel of Smalltalk/Squeak is all about (and it's something that was never quite completed in our Xerox PARC phase). The Japanese have a small word - ma - for "that which is in between" - perhaps the nearest English equivalent is "interstitial". The key in making great and growable systems is much more to design how its modules communicate rather than what their internal properties and behaviors should be. Think of the internet - to live, it (a) has to allow ...

“The big idea is messaging” – Alan Kay

AKKA



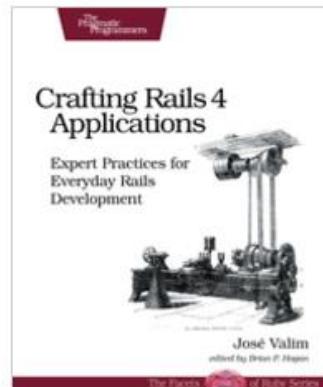
- “**Akka** is a free and open-source toolkit and runtime simplifying the construction of concurrent and distributed applications on the JVM. Akka supports multiple programming models for concurrency, but it emphasizes actor-based concurrency, with inspiration drawn from Erlang.” – Akka, Wikipedia
-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



JOSE VALIM



SAY HI TO JOSE VALIM



RUBY ON RAILS CORE
CONTRIBUTOR



I NEED “HIGH PERFORMANT SOFTWARE” - JOSE

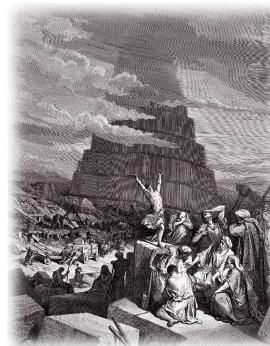
The
Pragmatic
Programmers

Seven Languages in Seven Weeks

A Pragmatic
Guide to
Learning
Programming
Languages

Bruce A. Tate

Edited by Jacquelyn Carter



IN LOVE WITH ERLANG

- He 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

- 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

- Tools – `elixir`, `elixirc`, `iex`
- Standard Library
- Mix – The comprehensive build system : Best practices, code organization, code generation
- ExUnit -

Stable ([download](#))

- [Elixir](#) - standard library
- [EEx](#) - templating library
- [ExUnit](#) - unit test library
- [IEx](#) - interactive shell
- [Logger](#) - built-in Logger
- [Mix](#) - build tool



ELIXIR IN ACTION...



AGENDA

- Distribution
- Iex
- :observer



ECOSYSTEM



THE POWER OF ELIXIR

- Metaprogramming
- Utilizing the Erlang VM



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.



CODE EXAMPLE



RESOURCES TO LEARN ELIXIR

