Manuel Meyer | Trivadis AG
www.manuelmeyer.net - @manumeyer1

22.09.2016

Concurrent WPF mit Akka.NET

Über mich

- Consultant & Trainer für .NET bei der Trivadis AG
- MVP für Visual Studio
- C# / XAML, Integration, Azure, Troubleshooting & Performance Management

Manuel Meyer

http://manuelmeyer.net

@manumeyer1

Microsoft® MVP Most Valuable Professional

Agenda

- History
- The Actor Model
- Akka.NET Principles
- Demo

Agenda

- **History**
- The Actor Model
- Akka.NET Principles
- Demo

Erlang

- Parallel
- Highly Available
- Fault-Tolerant
- Hot-Pluggable

- > COPL
- > Build MSSRTS.

Erlang

Prolog
Smalltalk
PLEX

F#
Clojure
Rust
Scala
Opa
Reia
Elixir
Dart
Akka

Erlang QuickSort

```
%% quicksort(List)
%% Sort a list of items
-module(quicksort).
-export([qsort/1]).

qsort([]) -> [];
qsort([Pivot|Rest]) ->
    qsort([ X || X <- Rest, X < Pivot]) ++ [Pivot]
++ qsort([ Y || Y <- Rest, Y >= Pivot]).
```

```
-module(ping_pong).
-export([ping/0, pong/0]).

ping() ->
    Receiver = spawn(ping_pong, pong, []),
    Receiver ! {self(), ping},
    receive
        pong ->
            pong
    end.

pong() ->
    receive
        {Sender, ping} ->
            Sender ! pong
    end.
```

Erlang Users

CouchDB

RabbitMQ

SimpleDB (AWS)

ejabberd

Cowboy, Ranch, Bullet, Sheriff

YAHOO!

GitHub

T··Mobile·

GMX

Akka.NET

Simple Concurrency & Distribution

Asynchronous and Distributed by design. High-level abstractions like Actors and FSM.

High Performance

50 million msg/sec on a single machine. Small memory footprint; ~2.5 million actors per GB of heap.

Resilient by Design

Write systems that self-heal. Remote and/or local supervisor hierarchies.

Elastic & Decentralized

Adaptive load balancing, routing, partitioning and configuration-driven remoting.

Extensible

Use Akka.NET Extensions to adapt Akka to fit your needs.

Open Source

Akka.NET is released under the Apache 2 license

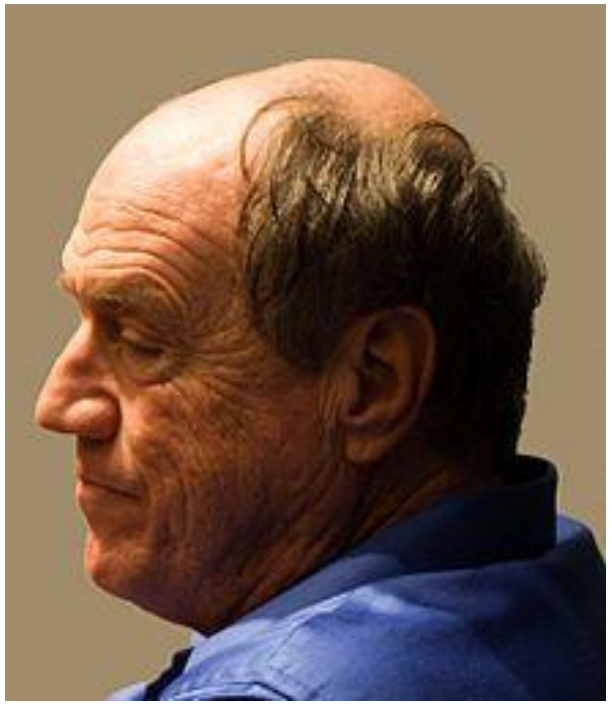
Agenda

- History
- **The Actor Model**
- Akka.NET Principles
- Demo

The Actor Model

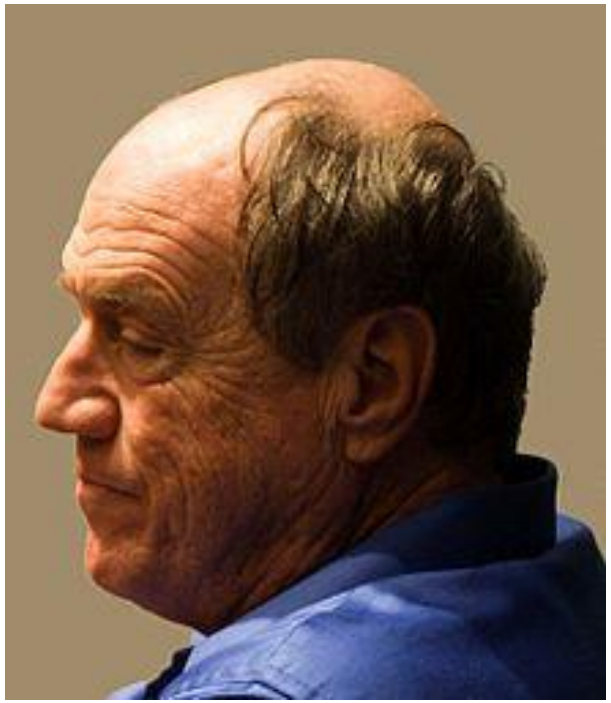
- Dr. Carl Hewitt (MIT) 1973

"A Mathematical Model of concurrent computation"

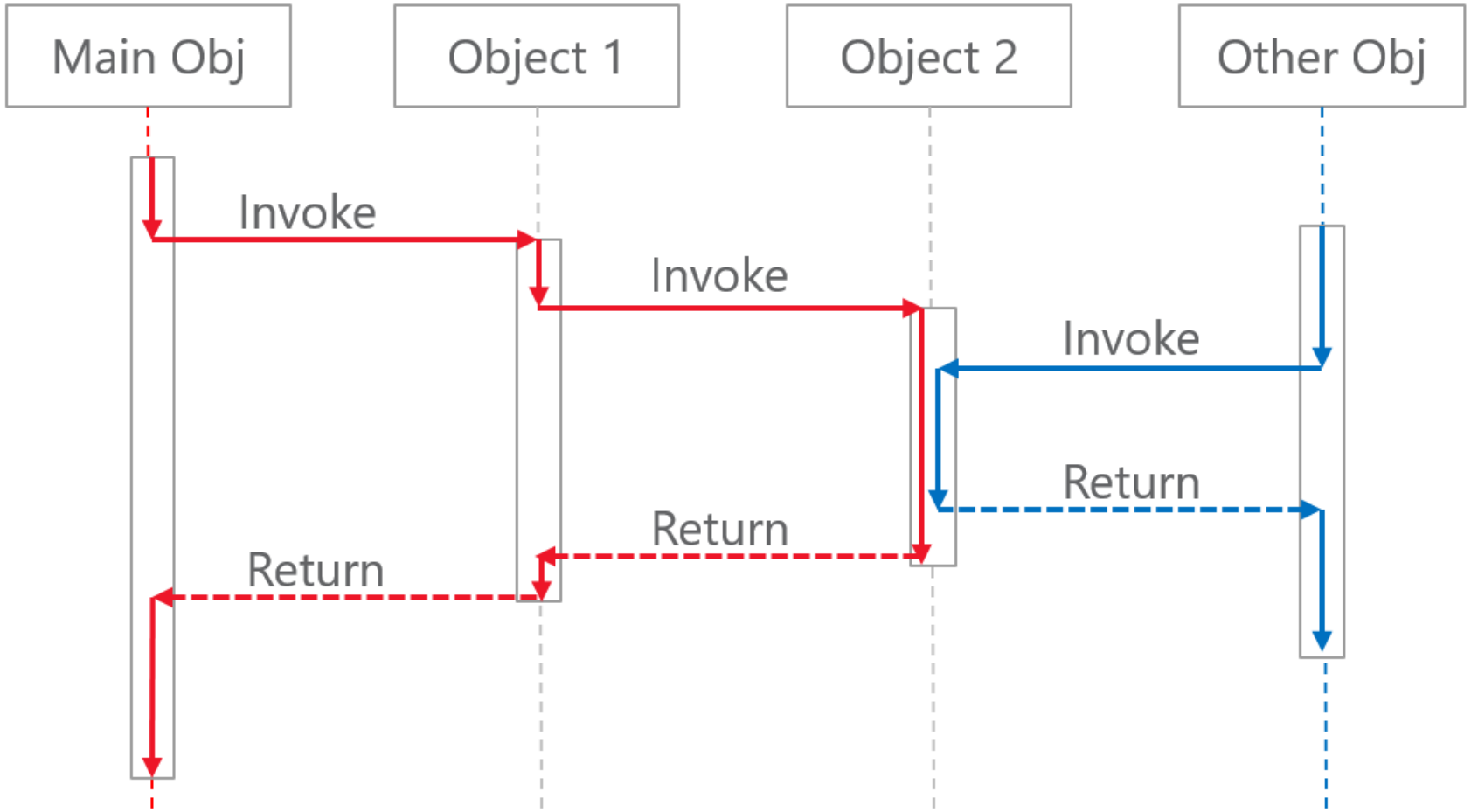


The Actor Model

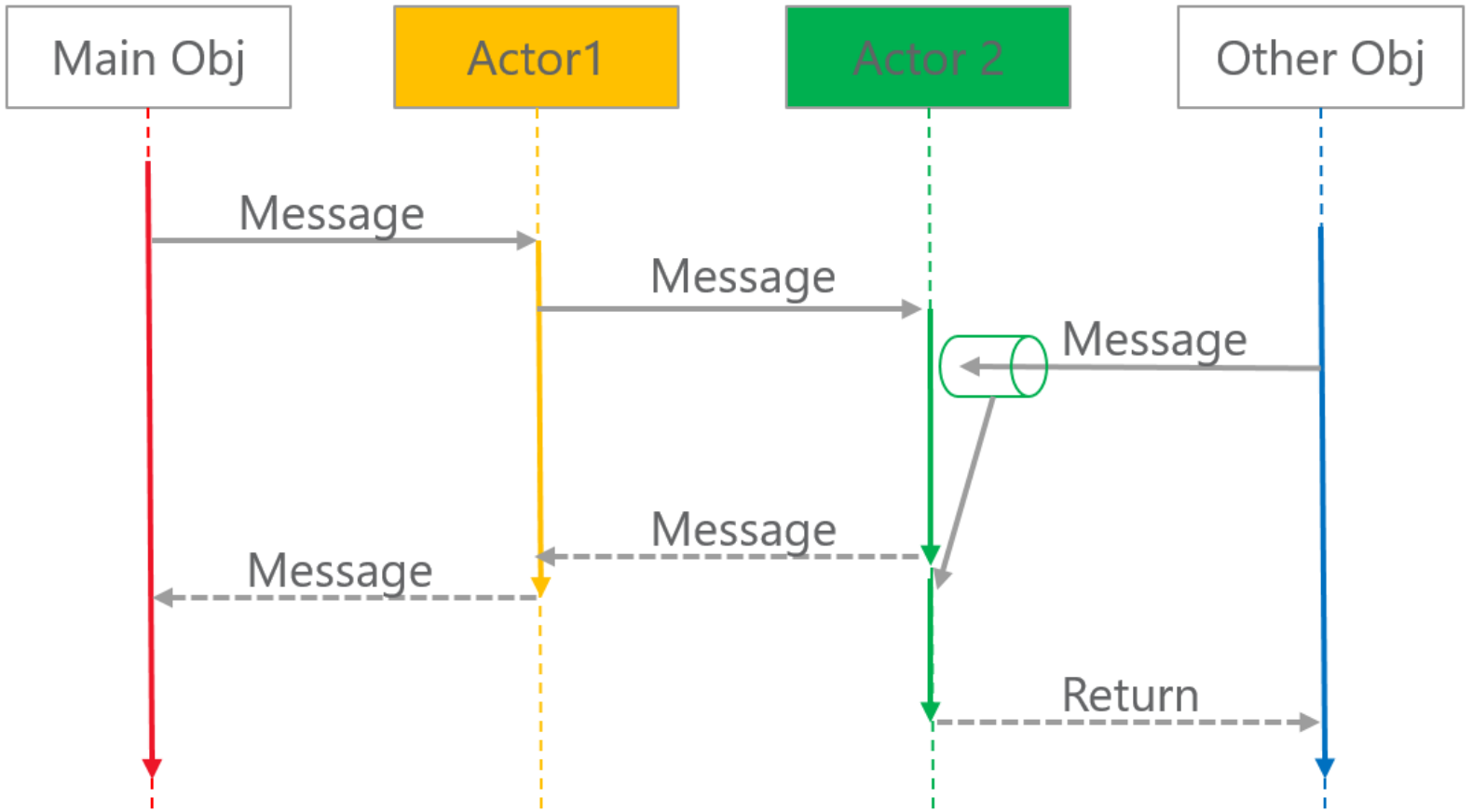
- „in the prospect of highly parallel computing machines with thousands of processors“
- Great fit for:
  - Cloud
  - Internet
  - Mobile
  - IoT
  - Reactive



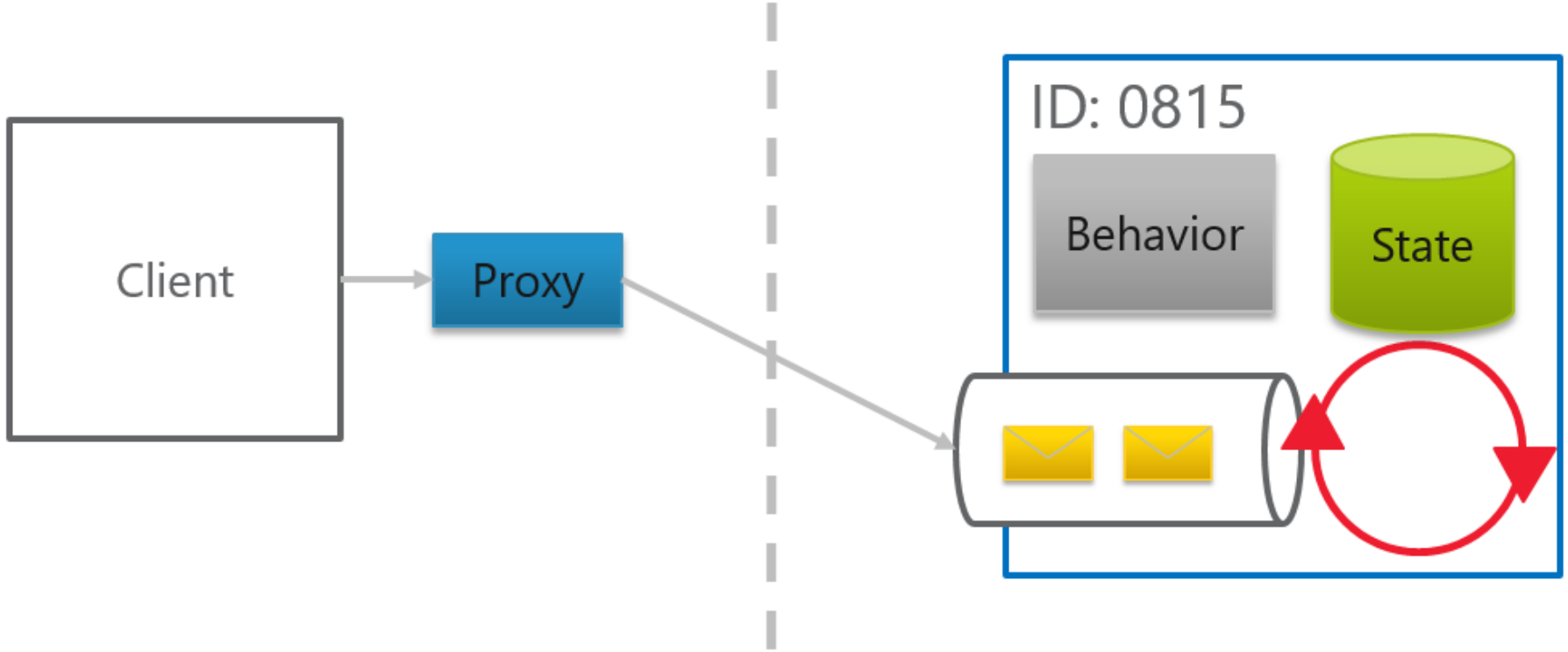
Actor Model


Main Obj
Object 1
Object 2
Other Obj
Invoke
Invoke
Invoke
Return
Return
Return

Actor Model


Main Obj
Actor1
Actor 2
Other Obj
Message
Message
Message
Message
Message
Return

Actor Model



Actor = (Computation, Storage, Communication)

Demo: Akka.NET

Agenda

- History
- The Actor Model
- **Akka.NET Principles**
- Demo

Akka Principles

- Everything is an Actor/Shared Nothing/Lightweight Actors
- Distributed by Default/Divide and Conquer
- Fault Tolerance/Supervision/Error-Kernel Pattern
- Loose Coupling/Location Transparency/Dynamics.

Akka Principles

- **Everything is an Actor/Shared Nothing/Lightweight Actors**
- Distributed by Default/Divide and Conquer
- Fault Tolerance/Supervision/Error-Kernel Pattern
- Loose Coupling/Location Transparency/Dynamics

Shared Nothing / Lightweight Actors

Actor System

```
ActorSystem.Create("MyActorSystem");
```

Actors

Props

```
Props.Create<ConsoleReaderActor>(
    SupervisorStrategy.DefaultStrategy);
```

Factories

```
MyActorSystem.ActorOf(props, "Sepp");
```

Messages (POCO)

```
actor1.Tell(startProcessingMessage);
var response = actor1.Ask(reportStatusMessage);
```

Shared Nothing / Lightweight Actors

- 3 Mio. Actors per GB of RAM
- Passive Actors.

Akka Principles

- Everything is an Actor/Shared Nothing/Lightweight Actors
- **Distributed by Default/Divide and Conquer**
- Fault Tolerance/Supervision/Error-Kernel Pattern
- Loose Coupling/Location Transparency/Dynamics

Distributed by Default / Divide & Conquer

1. Take a huge pile of work
2. Break it down until it is easy to deal with
3. Respond as needed.

Akka Principles

- Everything is an Actor/Shared Nothing/Lightweight Actors
- Distributed by Default/Divide and Conquer
- **Fault Tolerance/Supervision/Error-Kernel Pattern**
- Loose Coupling/Location Transparency/Dynamics

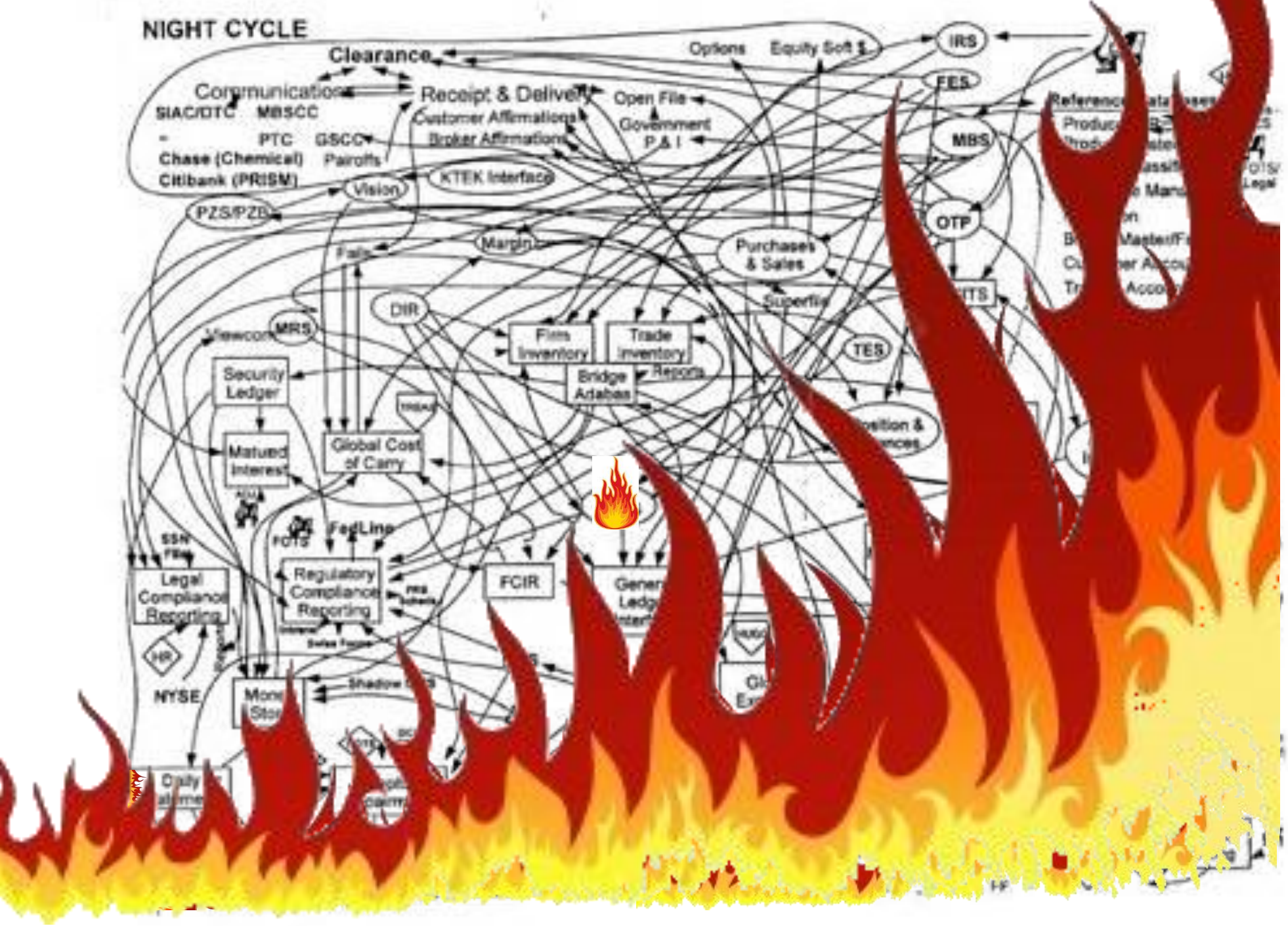
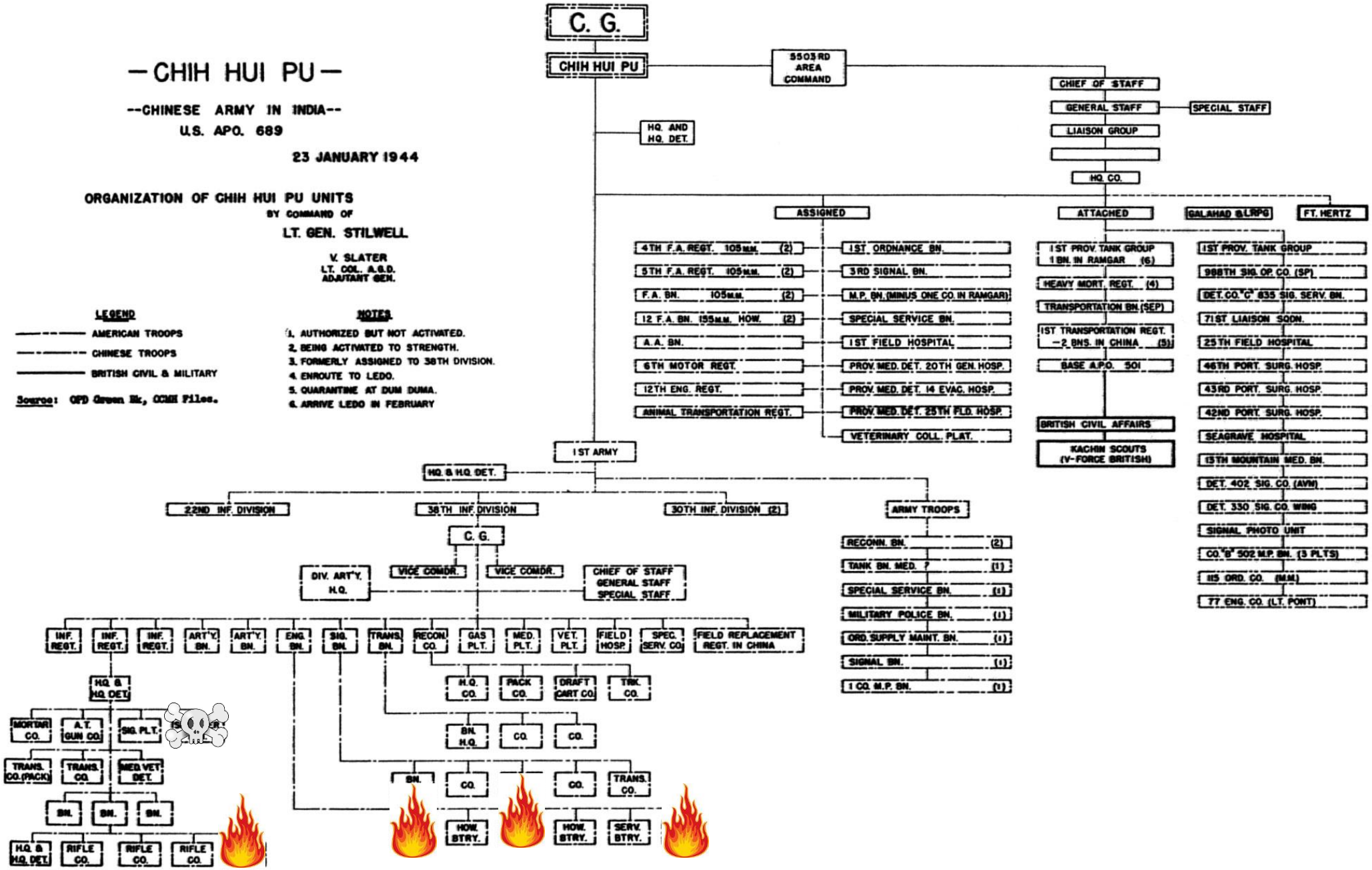
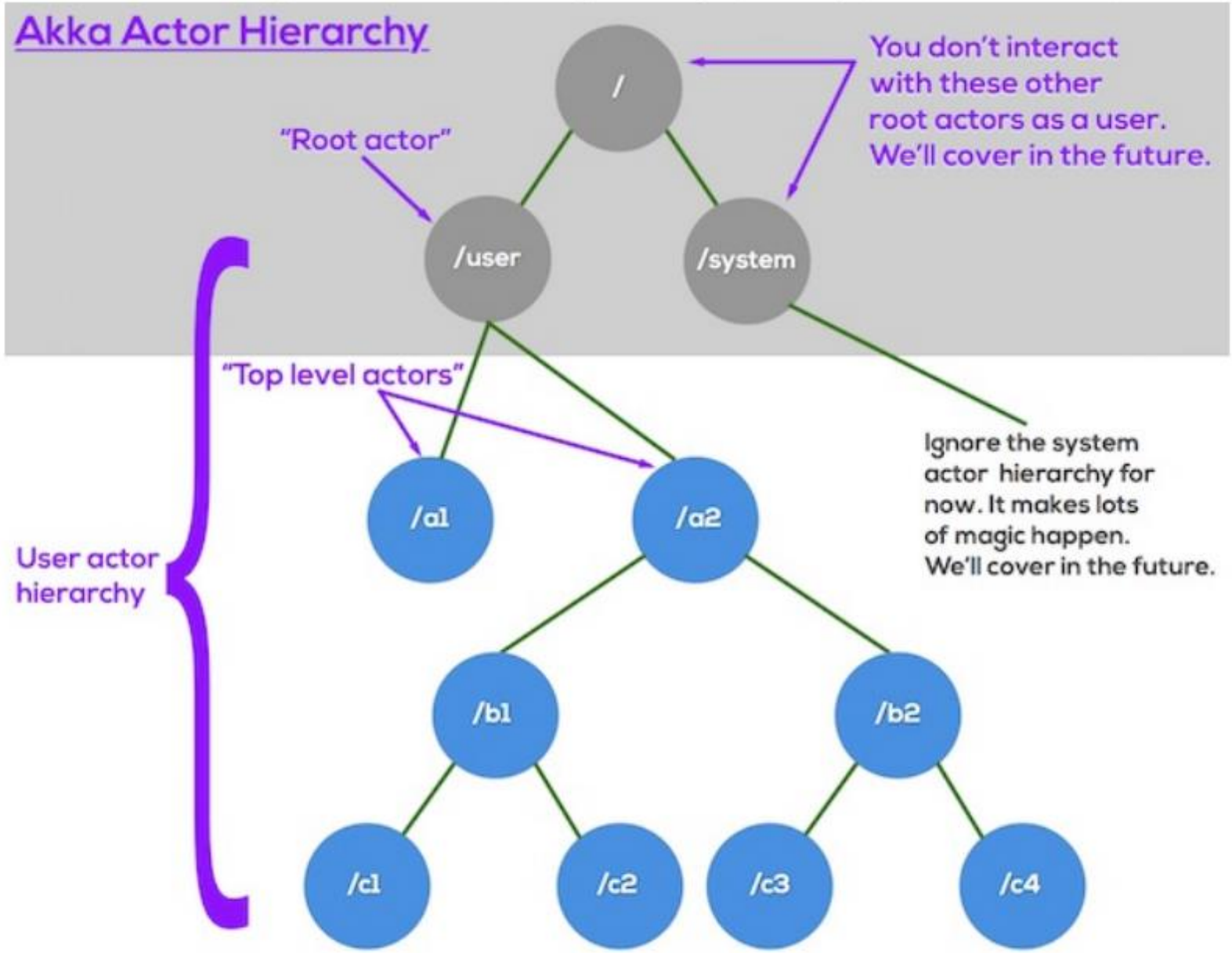

NIGHT CYCLE
Clearance
Communications
Receipt & Delivery
Options
Equity Soft $
IRS
FES
MBS
OTP
SIAC/DTC
MBSCC
PTC
GSCC
Chase (Chemical)
Payoffs
Citibank (PRISM)
Customer Affirmations
Broker Affirmations
Open File
Government P & I
KTEK Interface
Vision
PZS/PZB
Margin
Purchases & Sales
DIR
MRS
Firm Inventory
Trade Inventory
Repos
Bridge Arabes
TES
Security Ledger
Matured Interest
Global Cost of Carry
FedLine
FCIR
Legal Compliance Reporting
Regulatory Compliance Reporting
HR
NYSE

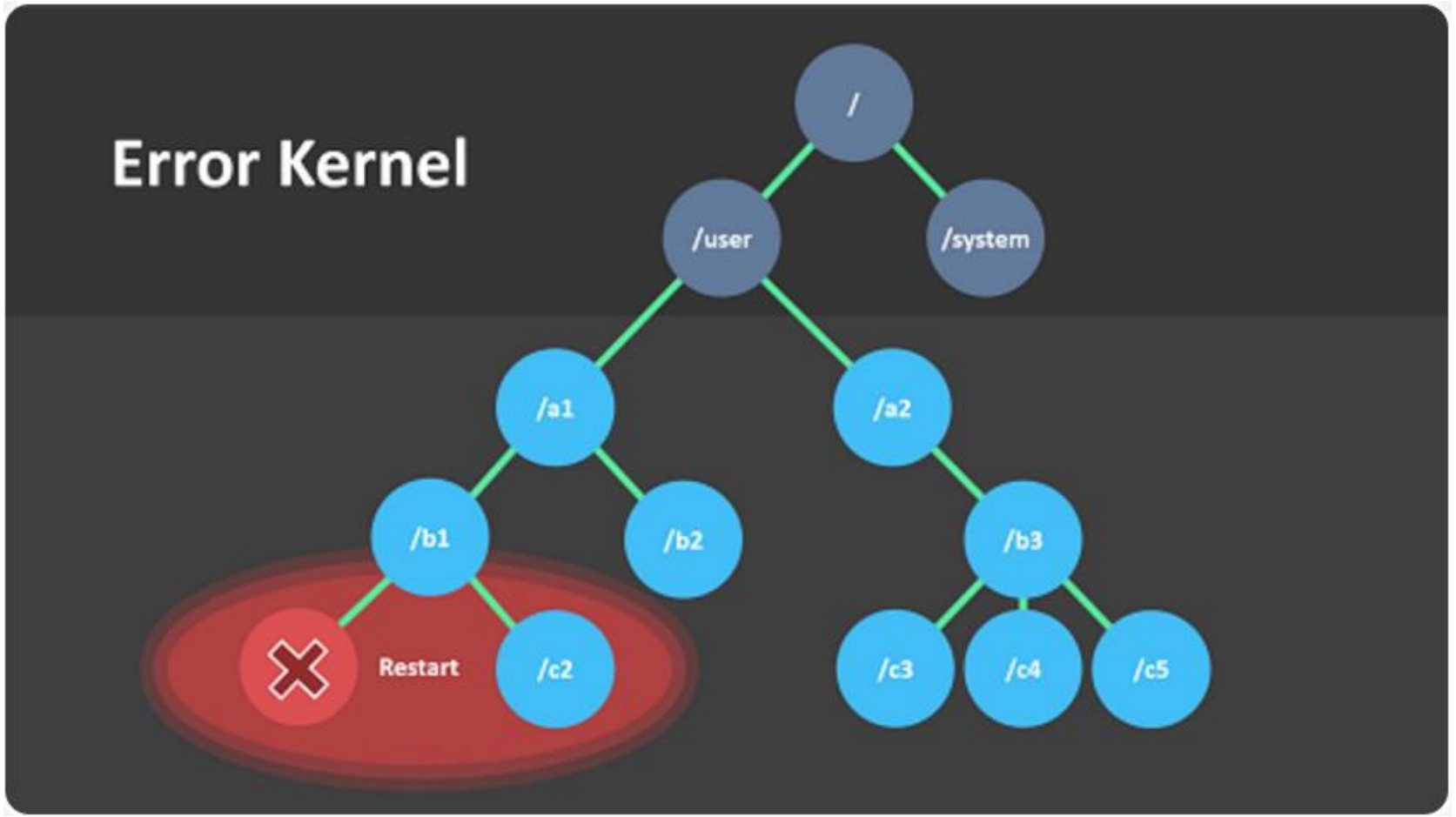
Chart 3—Chih Hui Pu





Source: Petabridge Bootcamp

Fault Tolerance / Error Kernel Pattern



Akka Principles

- Everything is an Actor/Shared Nothing/Lightweight Actors
- Distributed by Default/Divide and Conquer
- Fault Tolerance/Supervision/Error-Kernel Pattern
- **Loose Coupling/Location Transparency/Dynamics**

Loose Coupling / Location Transparency

IActorReference:

```
// Create an Actor!
IActorRef consoleWriterActor = MyActorSystem.ActorOf(Props.Create(()
    => new ConsoleWriterActor()), "consoleWriterActor");
```

ActorPath:

akka.tcp://MyActorSystem@LTMME:9001/user/actorName1

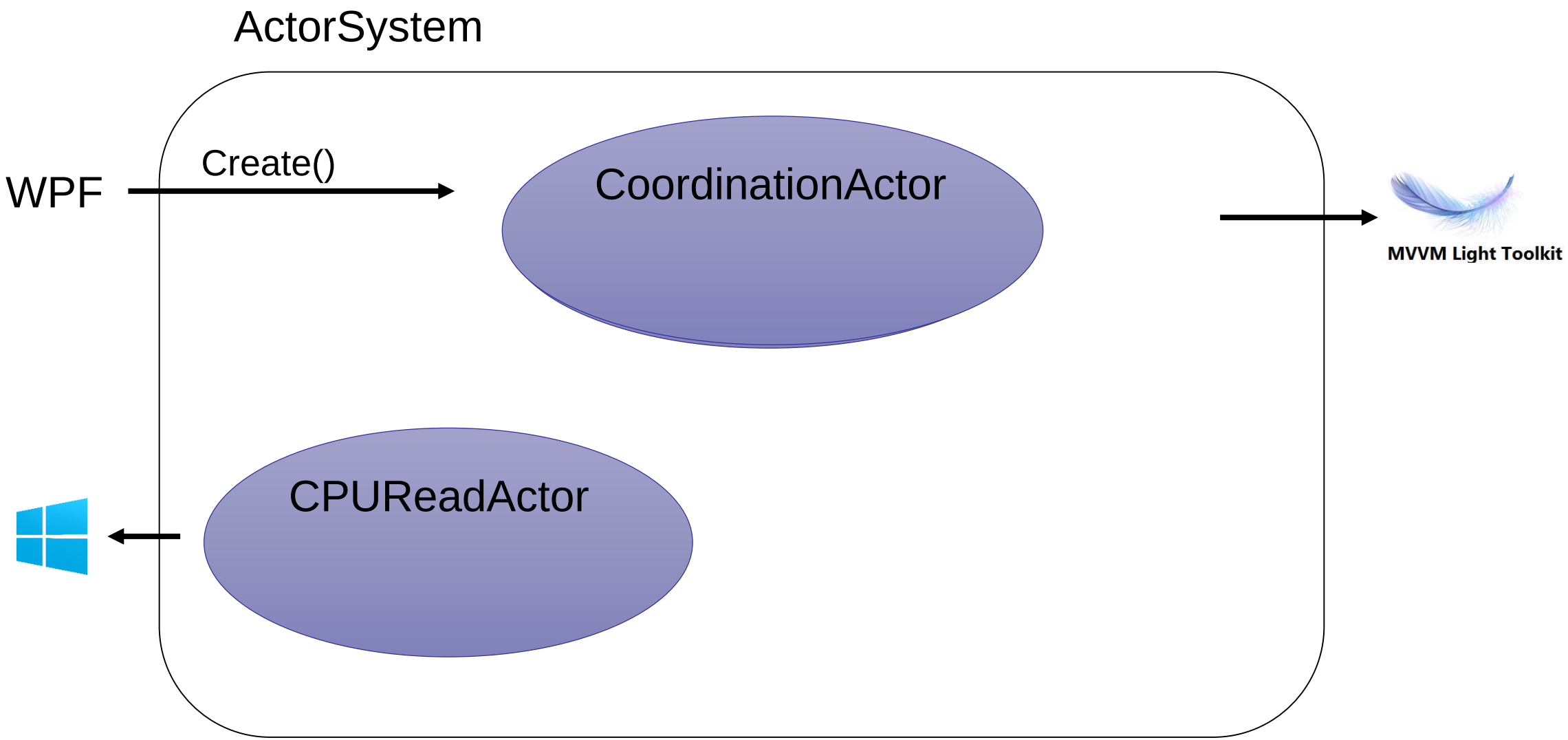
```
var x = MyActorSystem.ActorSelection(
    "akka.tcp://MyActorSystem@LTMME:9001/user/consoleWriterActor");
x.Tell("message");
```

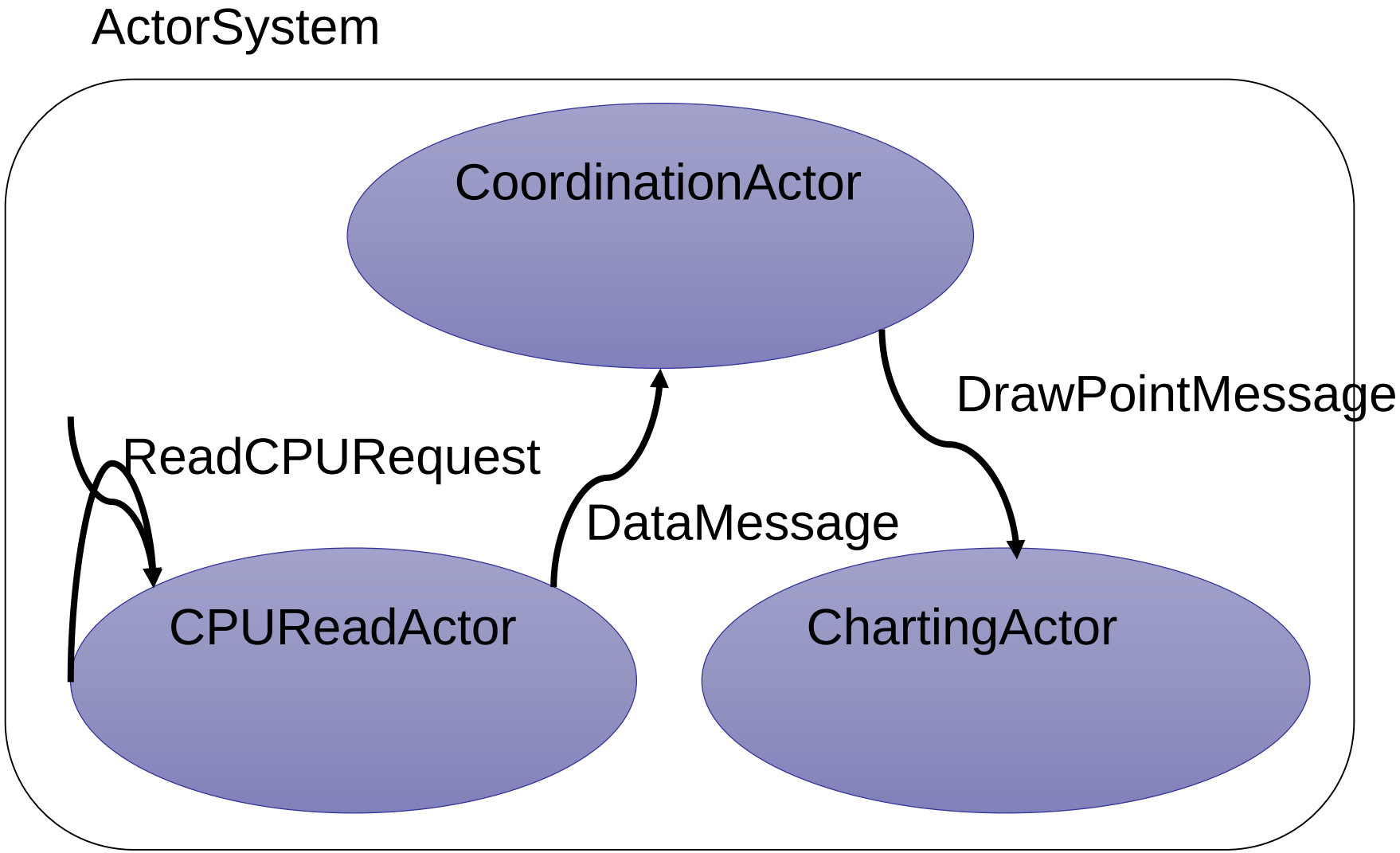
Agenda

- History
- The Actor Model
- Akka.NET Principles
- **Demo**

Demo

- Read CPU Data continuously
- Perform Calculations
- Give Data to Viewmodel


ActorSystem
WPF
Create()
CoordinationActor
MVVM Light Toolkit
CPUReadActor


ActorSystem
CoordinationActor
DrawPointMessage
ReadCPURequest
DataMessage
CPUReadActor
ChartingActor

Resources

- Akka.NET
  - http://getakka.net
- Petabridge Akka.NET Bootcamp
  - https://petabridge.com/bootcamp/
- Pluralsight (www.pluralsight.com)
  - Akka.NET Fundamentals
  - WPF, SPA, REST

Manuel Meyer | Trivadis AG
www.manuelmeyer.net
@manumeyer1

Concurrent WPF mit Akka.NET