Evolvix and HTCondor

What would it take to integrate

a

type system for computing tasks

and a

general-purpose programming language

Laurence Loewe

Wisconsin Institute for Discovery and Laboratory of Genetics

University of Wisconsin-Madison

2017 Condor Week, UW-Madison, Fluno Center, 2017-05-05

Update: MMv2_2017m09d19

Evolvix and HTCondor

What would it take to integrate

a

type system for computing tasks

and a

general-purpose programming language

or

How to make HTCondor Invisible

=

Usable

Laurence Loewe

Wisconsin Institute for Discovery and Laboratory of Genetics

University of Wisconsin-Madison

2017 Condor Week, UW-Madison, Fluno Center, 2017-05-05

Update: MMv2_2017m09d19

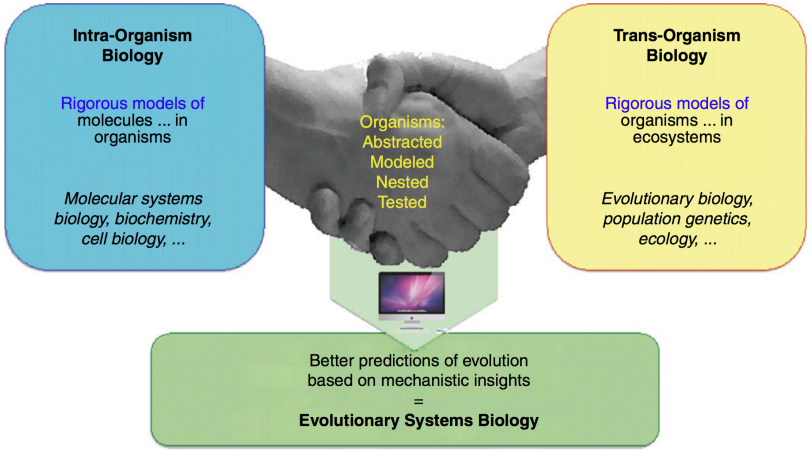
Type System for Compute Tasks

- **Why?**
  Cut inessential complexity in EvoSysBio research
- **How?**
  Flipped Programming Language Design
- **Assumptions?**
  What users and developers have to provide
- **Semantics + Syntax?**
  MockupModel defined using BEST Names
- **What next?**
  FeedbackFlow: discuss use cases, ambiguities, ...

Type System for Compute Tasks

- **Why?**
  Cut inessential complexity in EvoSysBio research
- **How?**
  Flipped Programming Language Design
- **Assumptions?**
  What users and developers have to provide
- **Semantics + Syntax?**
  MockupModel defined using BEST Names
- **What next?**
  FeedbackFlow: discuss use cases, ambiguities, ...

Why?



- **My Goal: Integrate diverse biological data**

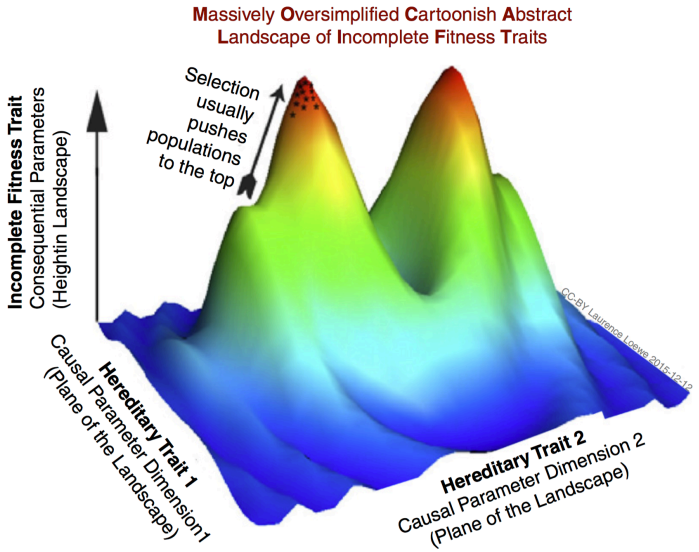
  to help develop real-world-informative

  Flightsimulators for Fitness landscapes

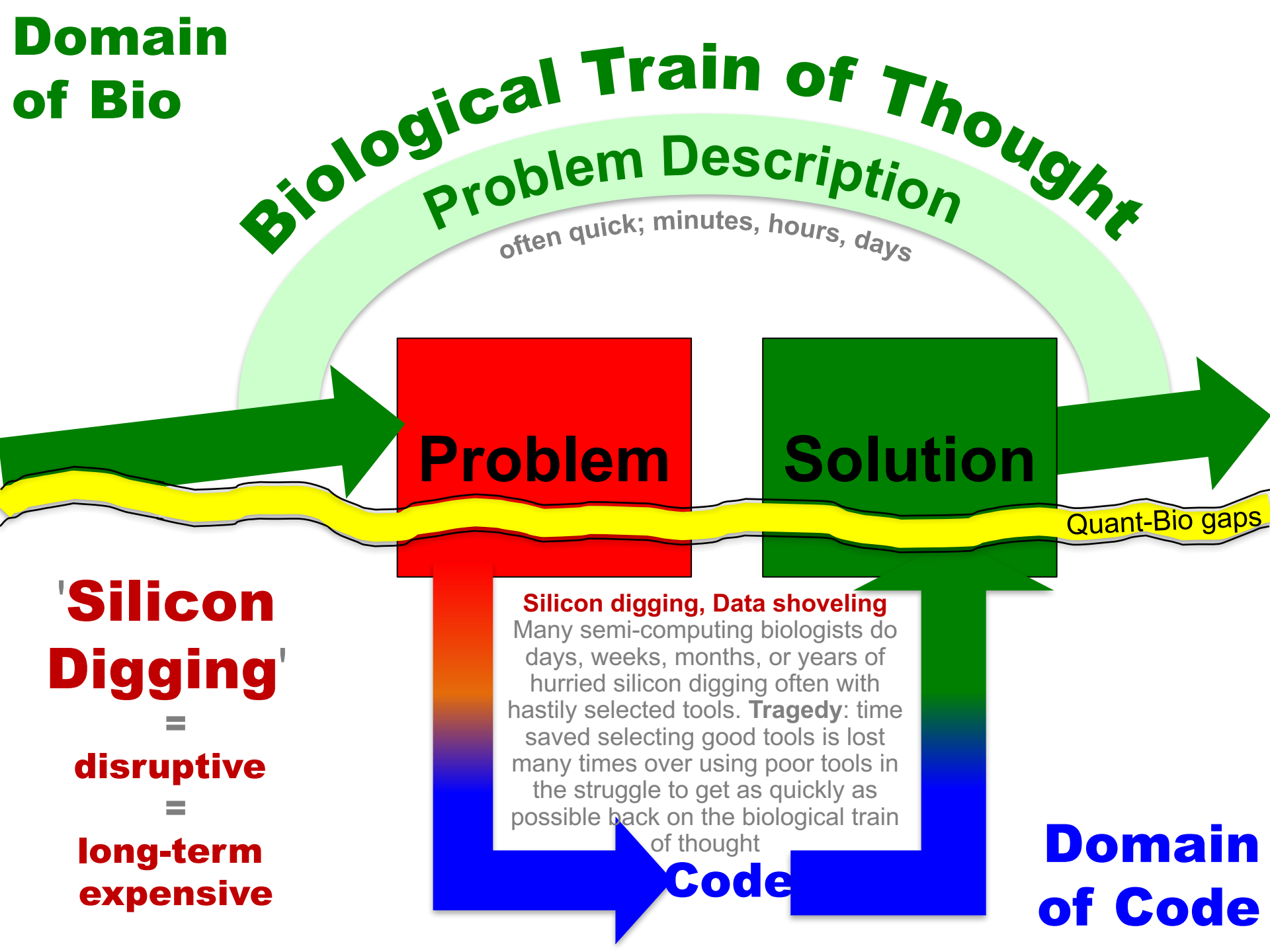
  - predicting cancer cell evolution in mouse, etc.

  - handle massively concurrent systems

- **EvoSysBio *requires***
  - ease-of-use
  - long-term stability
  - backwards compatibility



See: Loewe (2016) "Systems in Evolutionary System Biology" in *Encyclopedia of Evolutionary Biology*, Vol. 4 (ed. R Kliman) pp. 297-318, Academic Press (Elsevier), Oxford, UK. http://evolutionarysystemsbiology.org/pdf/Loewe-2016-evosysbio.pdf


Domain of Bio
Biological Train of Thought
Problem Description
often quick; minutes, hours, days
Problem
Solution
Quant-Bio gaps
'Silicon Digging'
=
disruptive
=
long-term
expensive
Silicon digging, Data shoveling
Many semi-computing biologists do days, weeks, months, or years of hurried silicon digging often with hastily selected tools. Tragedy: time saved selecting good tools is lost many times over using poor tools in the struggle to get as quickly as possible back on the biological train of thought
Code
Domain of Code

Imagine similar approaches when using cars ...

Cars

are inefficient and dangerous without

reliable **social contracts**
= precompiled decisions

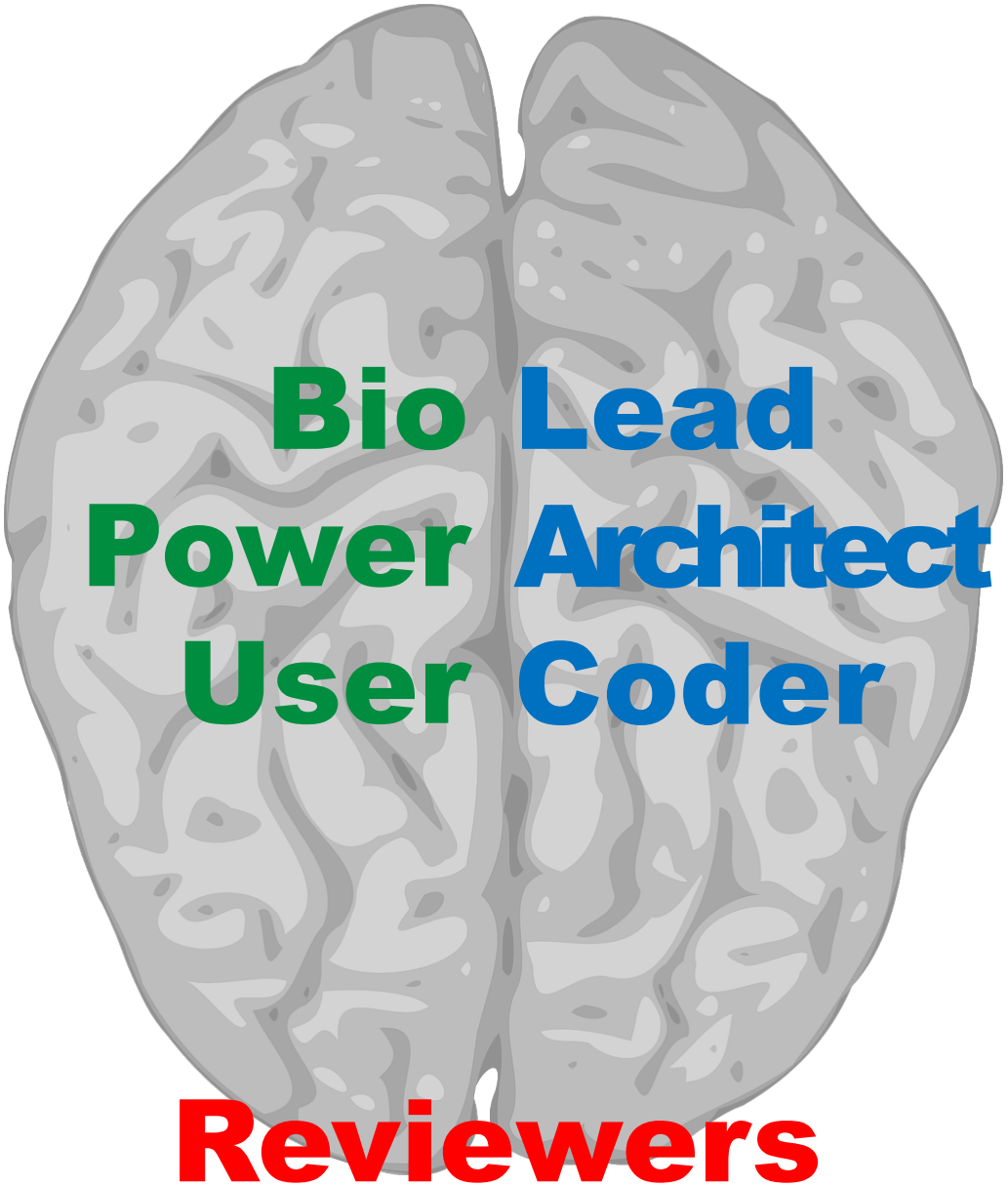
Renegotiating
on which side to drive,
one collaboration at a time...

➔ Develop
Reliable Standards

Type System for Compute Tasks

- **Why?**
  Cut inessential complexity in EvoSysBio research
- **How?**
  Flipped Programming Language Design
- **Assumptions?**
  What users and developers have to provide
- **Semantics + Syntax?**
  MockupModel defined using BEST Names
- **What next?**
  FeedbackFlow: discuss use cases, ambiguities, ...

Integrated Collaboration



Most new programming language cores are developed by 1-2 determined persons

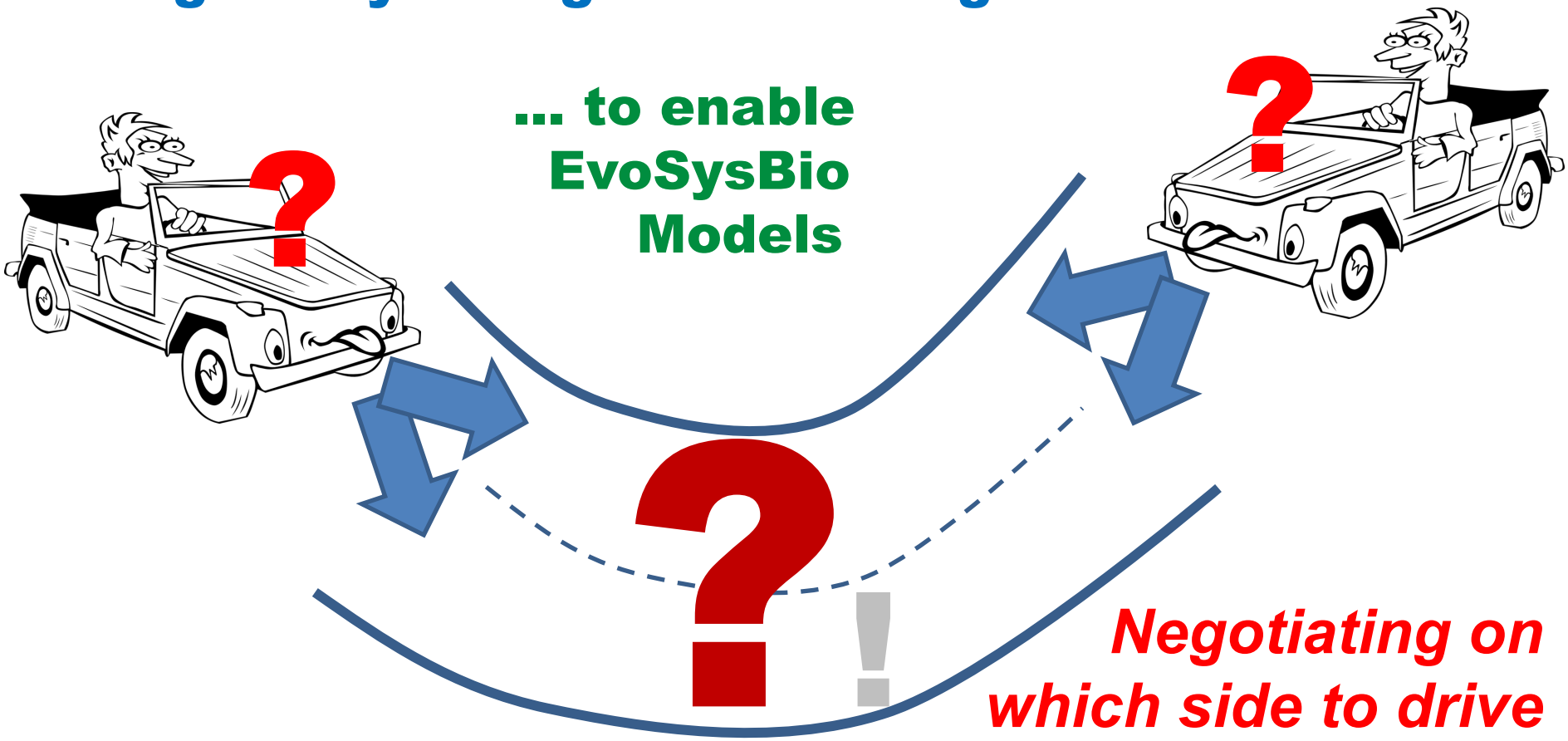
Advantages:

- Near instant 'user-developer conference calls'
- Decisions can be very fast if clear
- 20 years professional work in computational biology research are a great way to:
  - gather requirements
  - learn skills
  - recognize patterns

Caveats:

- Outside review is still *essential* to uncover blinds spots
- Decisions can be slow if requirement networks are complex

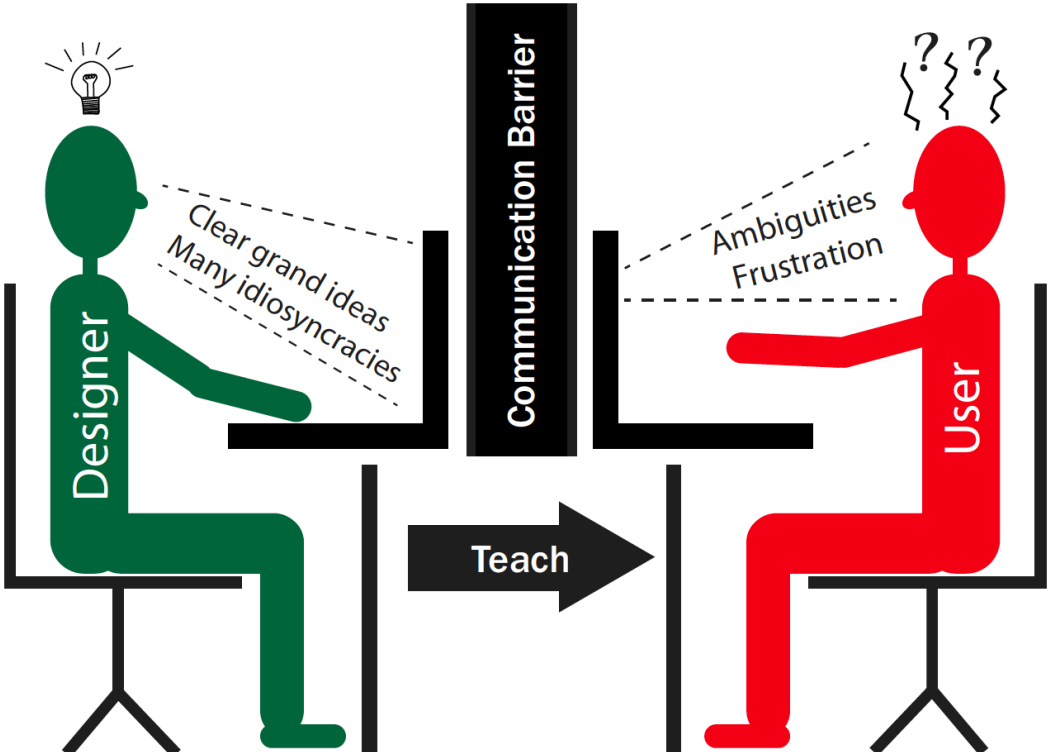
Evolvix Goal: **Make accurate modeling easier**
by architecting, implementing, and standardizing
the first general-purpose programming language
designed by biologists for biologists



Traditional Design Approaches will not work, because ...

Language designer

implements faster without interruptions from users, but is unaware of ambiguities, idiosyncrasies, and other problems that frustrate users ...



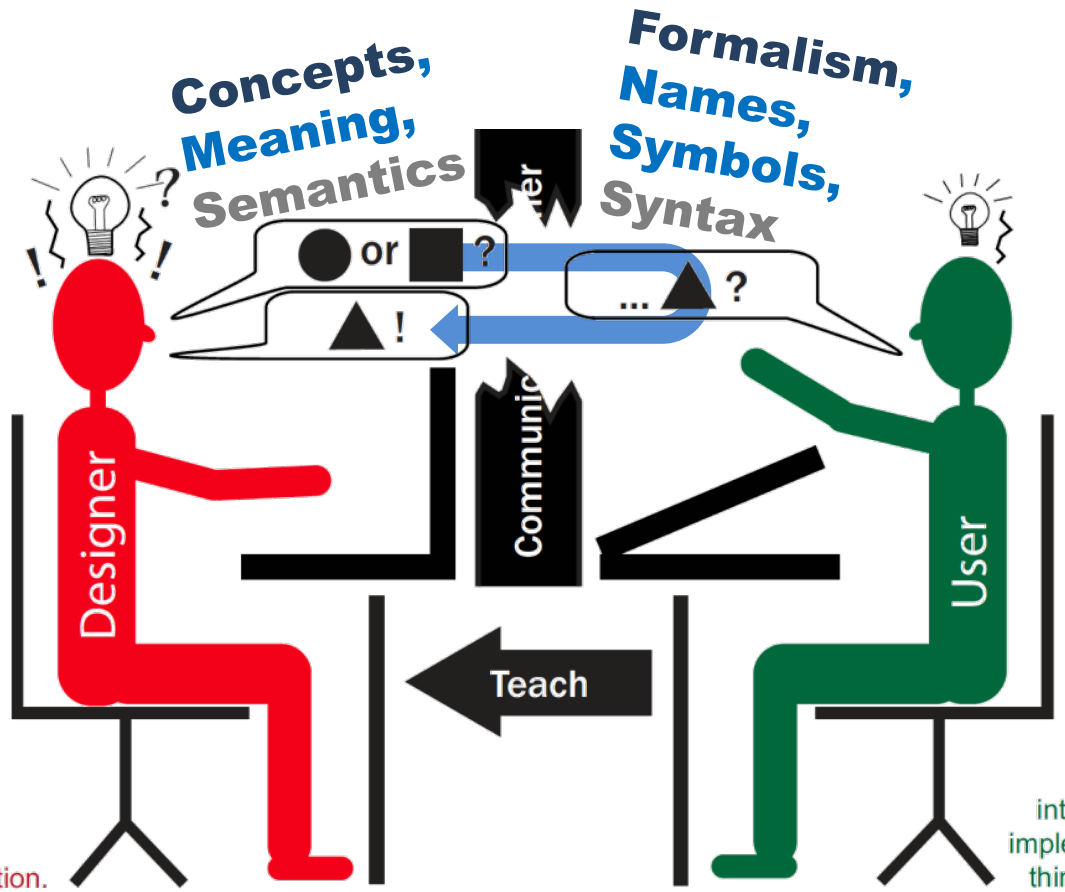
Users

have to learn what is given and suffer idiosyncracies, ambiguities, and inessential complexities they cannot change. Many eventually decide that "programming is not for me."

Loewe *et al.* (2016) "Evolvix BEST Names for semantic reproducibility across code2brain interfaces" *Ann. N.Y.Acad.Sci. 1387:124-144* http://dx.doi.org/10.1111/nyas.13192 ; http://evolvix.org/naming ; http://evolvix.org/post

… good editing needs *too* many iterations,
Clarity is a must *before* Implementation!

Language designer
is also a user, familiar with user problems, and has developed enough language design details so users can detect confusing syntax when discussing example code.
Reviewing clarity before implementation lock-in is the essence of the "flip": a language designer learns from users and removes many idiosyncrasies before implementation.
The curse of knowledge will inevitably trip up even the best designers unless they talk to potential users of their language. Users must bring a remarkable amount of patience for such work!

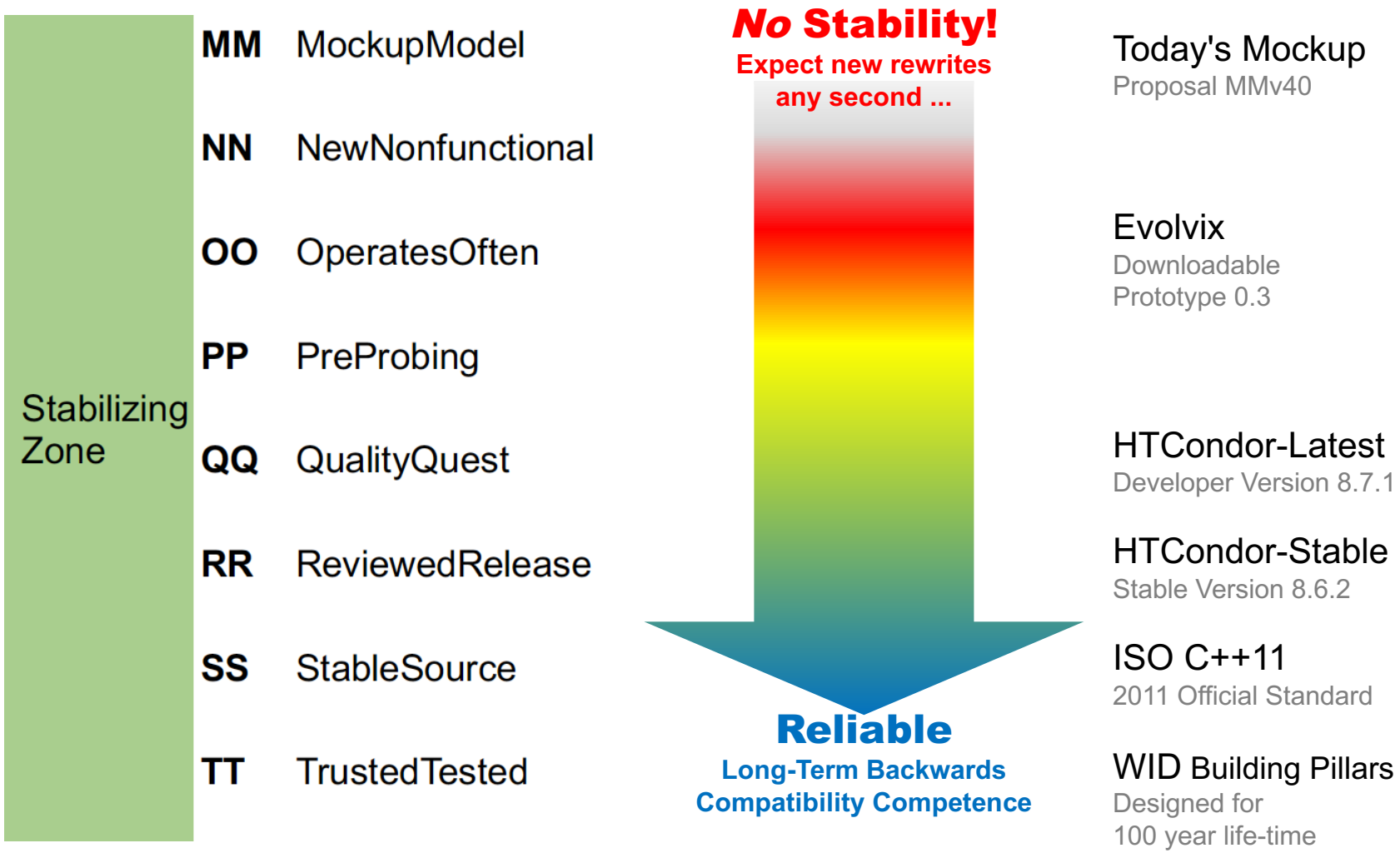


Users
do not have to suffer idiosyncracies, ambiguities, and inessential complexities over the long term if a few of them are willing to help the language designer in removing as many problems as possible from the language by sharing their confusions, preferences, interpretations, and ideas before implementation starts (when many things can no longer be changed except by implementing a new language).

Flipped programming language design

Loewe *et al.* (2016) "Evolvix BEST Names for semantic reproducibility across code2brain interfaces" *Ann. N.Y.Acad.Sci. 1387:124-144*
http://dx.doi.org/10.1111/nyas.13192 ; http://evolvix.org/naming ; http://evolvix.org/post

StabilityLevels Measure Progress of Variants on their way to Long-Term Stability

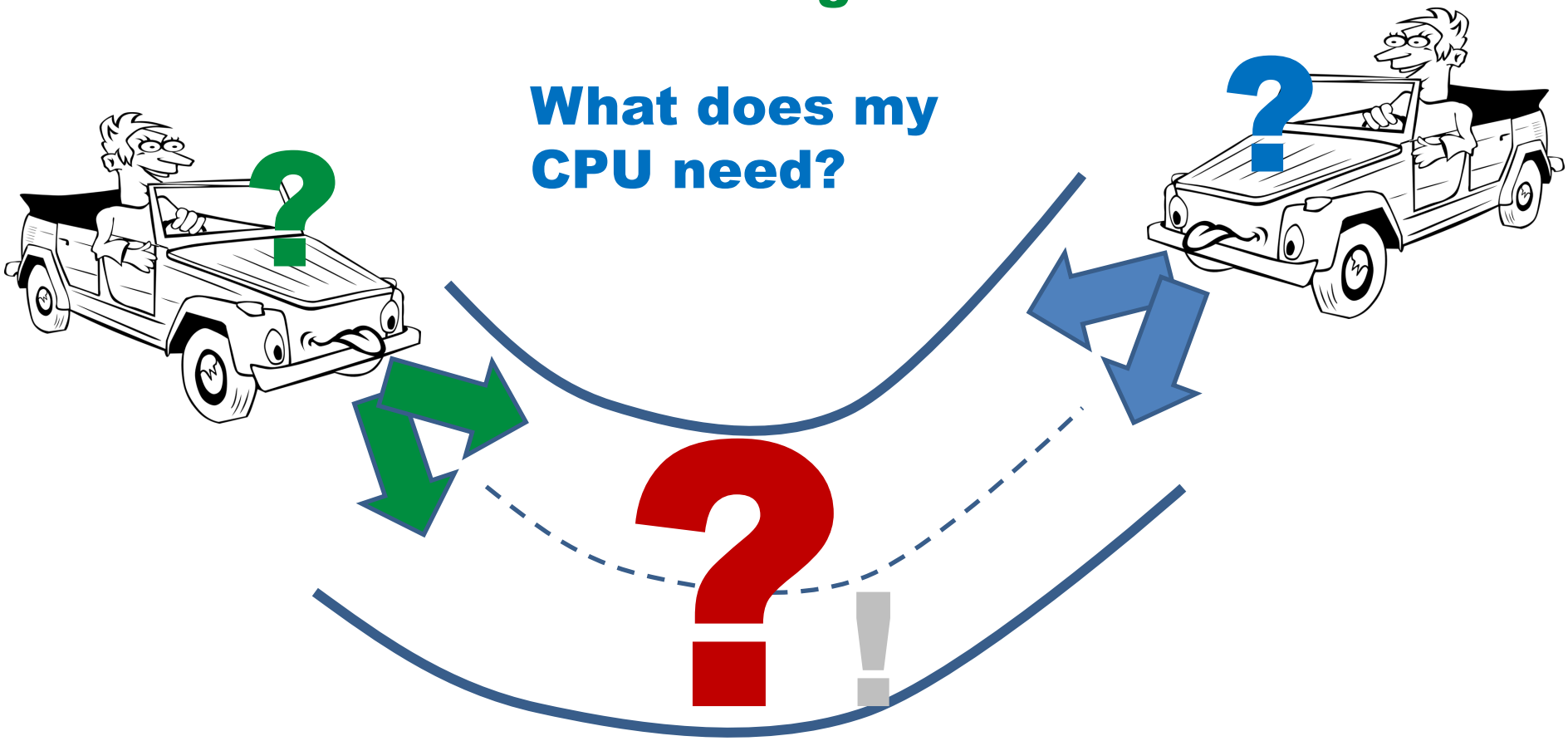


This *StabilizingZone* is part of the POST system developed for Evolvix; see Loewe *et al.* (2016) "Evolvix BEST Names for semantic reproducibility across code2brain interfaces" *Ann. N.Y.Acad.Sci. 1387:124-144* http://dx.doi.org/10.1111/nyas.13192 ; http://evolvix.org/post

Lets Negotiate ...

What do I need as a Biologist?

What does my CPU need?



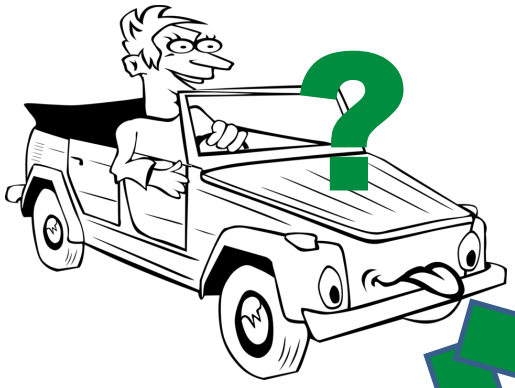
Type System for Compute Tasks

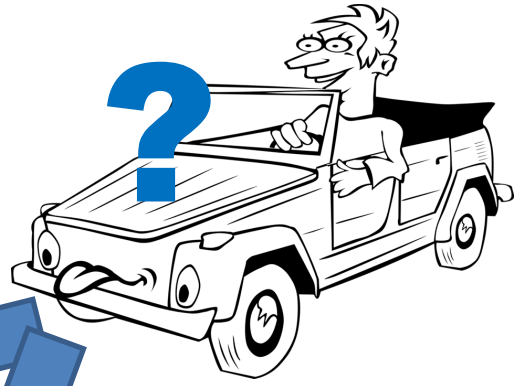
- **Why?**
  Cut inessential complexity in EvoSysBio research
- **How?**
  Flipped Programming Language Design
- **Assumptions?**
  What users and developers have to provide
- **Semantics + Syntax?**
  MockupModel defined using BEST Names
- **What next?**
  FeedbackFlow: discuss use cases, ambiguities, ...

Lets Negotiate ...

What do I need as a Biologist?

What does my CPU need?

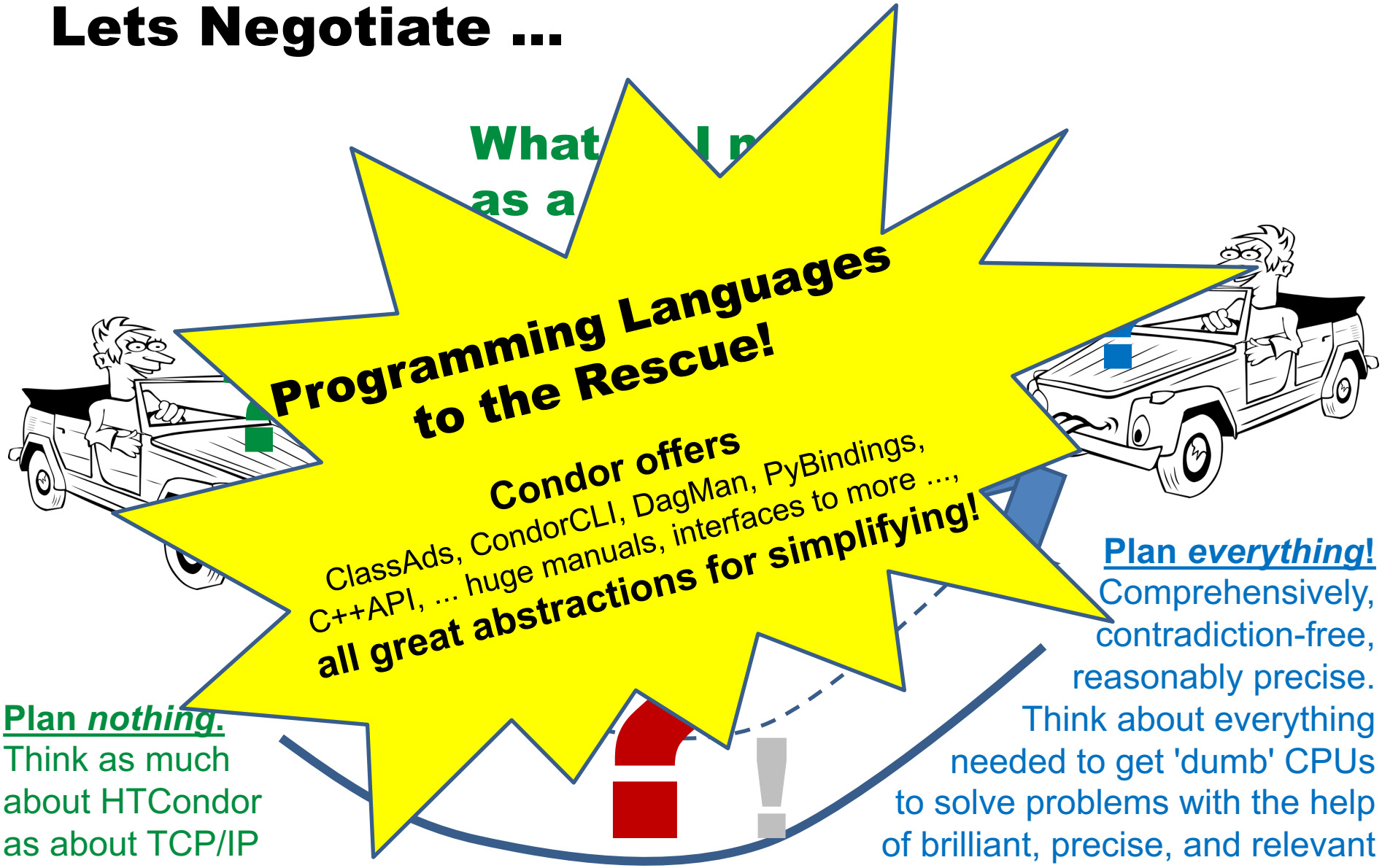




Plan *nothing*.
Think as much about HTCondor as about TCP/IP or ignition parameters in a car engine while starting:
Think *nothing*. Remember *nothing*.

Plan *everything*!
Comprehensively, contradiction-free, reasonably precise. Think about everything needed to get 'dumb' CPUs to solve problems with the help of brilliant, precise, and relevant instructions (may need reconfiguring and recompiling the OS kernel though)
Think about *everything*! **Remember *everything*!**

Lets Negotiate ...



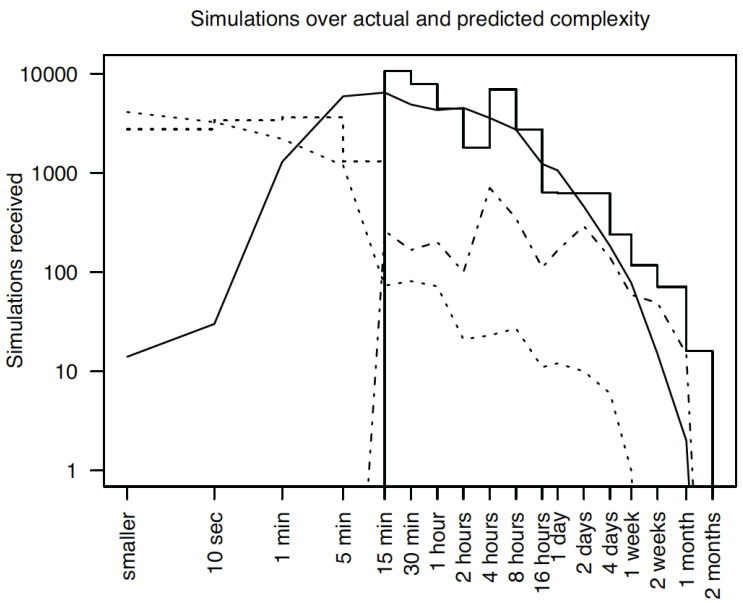
Plan *nothing*.
Think as much
about HTCondor
as about TCP/IP
or ignition parameters
in a car engine while starting:
Think *nothing*. Remember *nothing*.

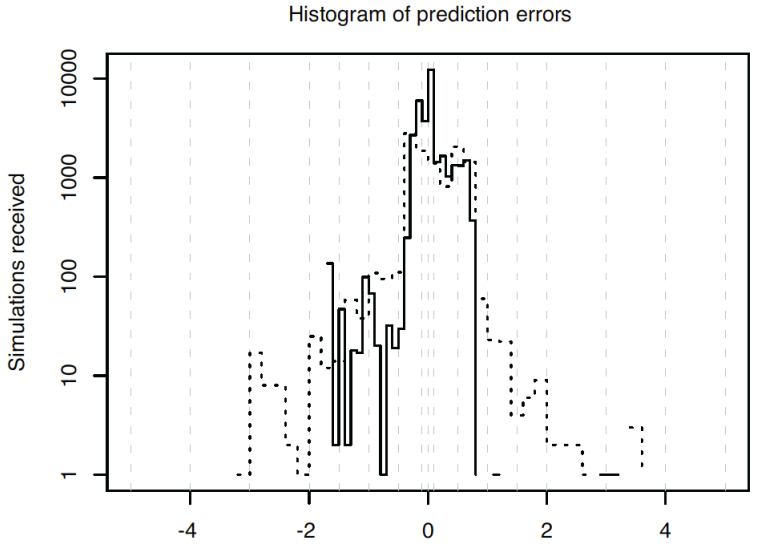
Plan *everything*!
Comprehensively,
contradiction-free,
reasonably precise.
Think about everything
needed to get 'dumb' CPUs
to solve problems with the help
of brilliant, precise, and relevant
instructions (may need reconfiguring
and recompiling the OS kernel though)
Think about *everything*! **Remember *everything*!**

Extreme Diversity in Biology ...

Millions of species, >10^{30} bacteria, many millions of years: generate about all exceptions imaginable. *And then some.*

... generates high compute diversity

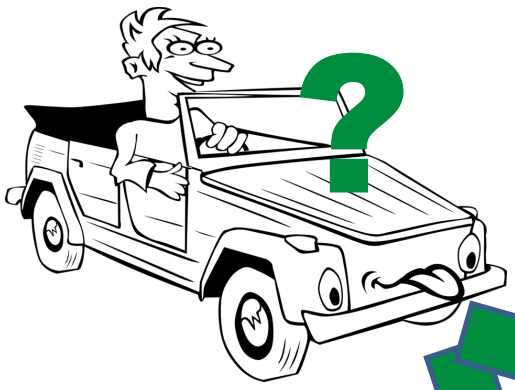




Loewe, L. Evolution@home: observations on participant choice, work unit variation and low-effort global computing. *Software Practice & Experience* **37**, 1289-1318 (2007).

Lets Negotiate ...

What do I need as a biologist



Plan *nothing*.

Do Biology!

Remember *nothing*.

Type System for Compute Tasks

- **Why?**
  Cut inessential complexity in EvoSysBio research
- **How?**
  Flipped Programming Language Design
- **Assumptions?**
  What users and developers have to provide
- **Semantics + Syntax?**
  MockupModel defined using BEST Names
- **What next?**
  FeedbackFlow: discuss use cases, ambiguities, ...

Draft Evolvix SystemSetup

Proposed formalism for storing configurations and passwords at secure locations

```
Evolvix SystemSetup .NickName_of_my_LaptopSetup {

    %Local_Owner_Info                   is .MyNickName {...Details...}
    %Local_Computer_TaskFast            is .MyNickName {Limits...}
    %Local_Computer_TaskSlow            is .MyNickName {Limits...}

    %Distributed_Computer_Door          is .MyNickName {Keys...  }
    %Distributed_Computer_TaskFast      is .MyNickName {Limits...}
    %Distributed_Computer_TaskSlow      is .MyNickName {Limits...}

    %Cloud_Computer_Door                is .MyNickName {Keys...  }
    %Cloud_Computer_TaskFast            is .MyNickName {Limits...}
    %Cloud_Computer_TaskSlow            is .MyNickName {Limits...}

}
```

Syntax Review Notes
Potential contrasts:

Tasks_Urgent Tasks_Later	
Tasks_Urgent Longer	Keys vs Passwords
Task_Faster-Slower	Task vs Tasks
Task_Fast-Slow	Door vs Access, Permit

Draft Evolvix GridSearch

Proposed formalism for systematic searches generating results along regular arrays

```
GridSearch .MyGridName    (
    ModelStructures
    (
        .MyModelA ,          .MyModelB ,    .MyModelC
    )
    ModelVariantDimensions
    (
                .MyDim1    (   .MyValues1              ),
                .MyDim2    (   .MyValues2              )
    )
    Query   (    .MyTimeSeriesDefName )
    Task    Simulate Deterministically Until    Time = 1000
)
$Where ( .MyNoteBookFast $Limits ( .MyCPUtime,    .MyRAMsize    ) )
$Where ( .MyCloudSlow    $Limits ( .MyCloudTime, .MyCloudSize ) )
```

Syntax Review Notes:
ModelStructures (elegant order implied) vs ModelStructure Set (unordered?)
vs ModelStructure Sequence (explicit order); similarly with ModelVariant Dimension Sequence

Importance of flexible Limits

For Task properties like CPU time, RAM used, Disk use, Bandwidth

Brief	Explicit	Summarizing	MatchResult
MaxHi	Upper Must Limit	Tasks using more: Kill, *Error*	***KO***
MaxLo	Upper Goal Limit	Tasks using more: End, *Warn*	*OKO*
		Tasks *should* use < MaxLo	OK
MidHi	Upper Aim Limit	Tasks try to schedule lower	OK
MidMid	Middle Aim Label	Easy referral to Tasks in this class	OK
MidLo	Lower Aim Limit	Tasks try to schedule higher	OK
		Tasks *should* use > MaxLo	OK
MinHi	Lower Goal Limit	Tasks using less: End, *Warn*	*OKO*
MinLo	Lower Must Limit	Tasks using less: End, *Error*	***KO***

Syntax Review Notes:
Must = OK; Goal vs Shall vs Should vs Planned vs Aim;
Ideal vs Terrifying vs Best vs Shall vs May vs Maybe

Type System for Compute Tasks

- **Why?**
  Cut inessential complexity in EvoSysBio research
- **How?**
  Flipped Programming Language Design
- **Assumptions?**
  What users and developers have to provide
- **Semantics + Syntax?**
  MockupModel defined using BEST Names
- **What next?**
  FeedbackFlow: discuss use cases, ambiguities, ...

FeedbackFlow
FF

more Use-Cases that
detect Ambiguities by
asking Domain Experts and
diverse Usability Experts for help

to review MMv2_2017m09d19 or newer variants.

Thank You for your attention!

You just completed your instant and subconscious 'gut-level' pre-evaluation of the concepts and formalisms presented here. Please consider contributing your anonymous unfiltered rants and/or more refined comments to the **FeedbackFlow** collected at:

http://evolvix.org/FF

http://evolvix.org/concept/tasktype

Thanks to Seth Keel and other Evolvix Thinkers for commenting
Thanks to NSF ABI Award 1149123 for Funding