

SLIM's software design principles

Tim Lin

Observations

A lot of our techniques rely on *seeing* the whole prestack dataset

non-separable optimizations,
multidimensional transforms,
randomly (re)drawn subsets ...

Observations

A lot of our techniques rely on
seeing the whole prestack dataset

EXPENSIVE

BIG

LOW-LEVEL IMPLEMENTATION

non-separable optimizations,

multidimensional transforms,

randomly (re)drawn subsets ...

but...

A lot of our techniques rely on *seeing* the whole prestack dataset

... but only working with parts at a time (under transposes)

but...

Conservation of Complexity
in terms of how much a student can handle

Physics

Math

Signal
Processing



STUDENT

Optimization

Computer
Architecture

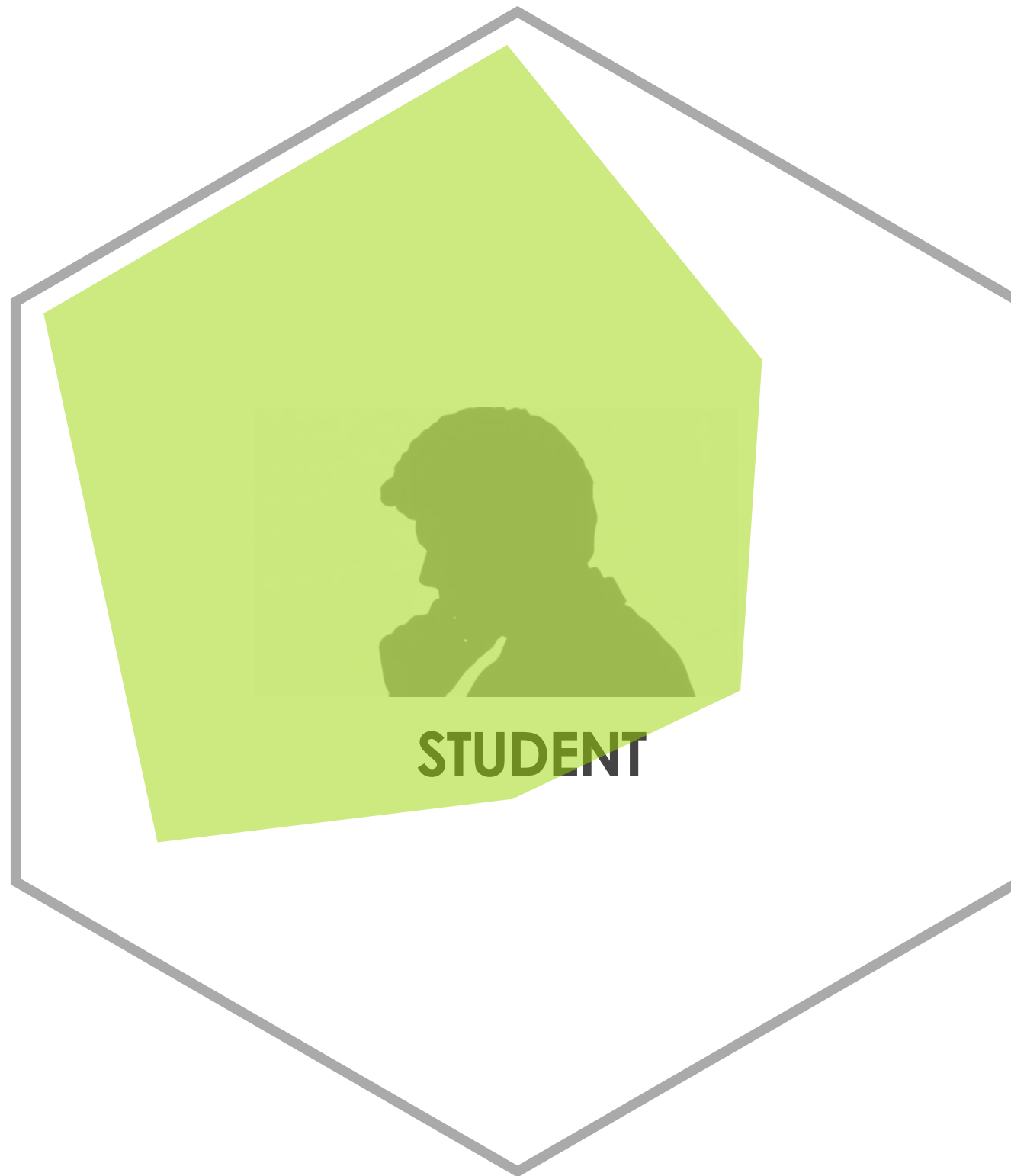
Programming

**theoretically
inclined
student**

Physics

Math

Signal
Processing



STUDENT

Optimization

Computer
Architecture

Programming

**computationally
inclined
student**

Physics

Math

Signal
Processing

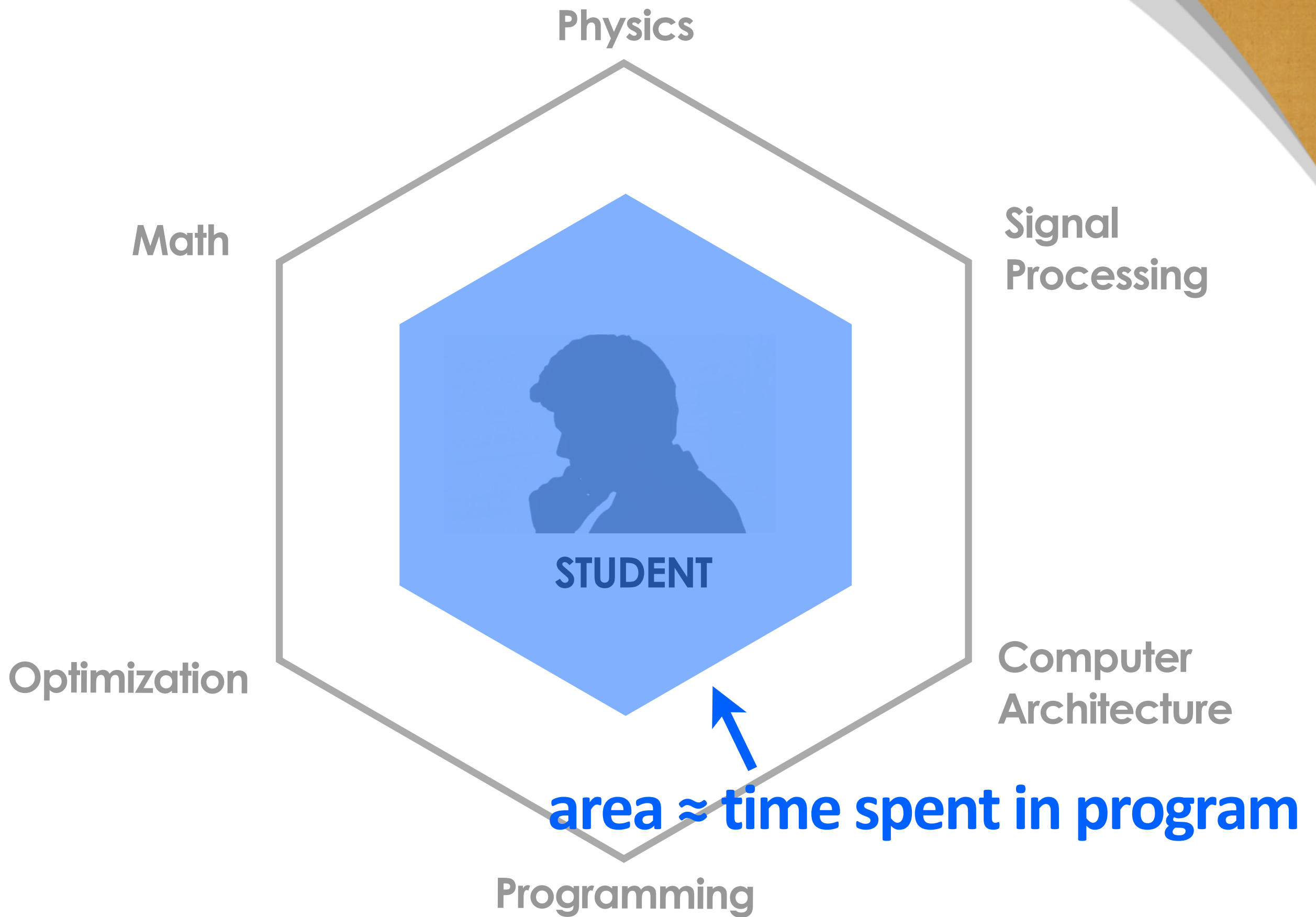


STUDENT

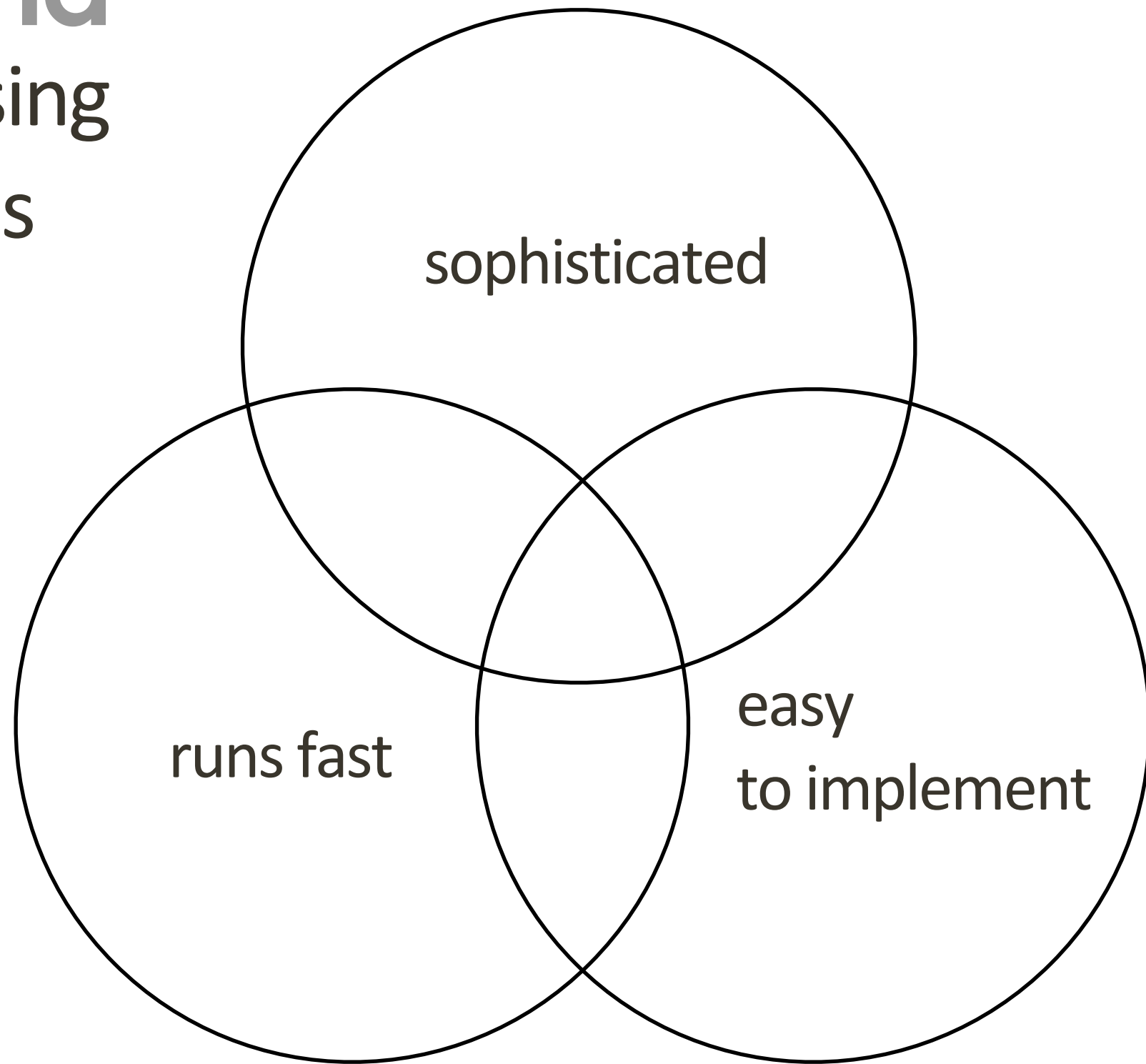
Optimization

Computer
Architecture

Programming



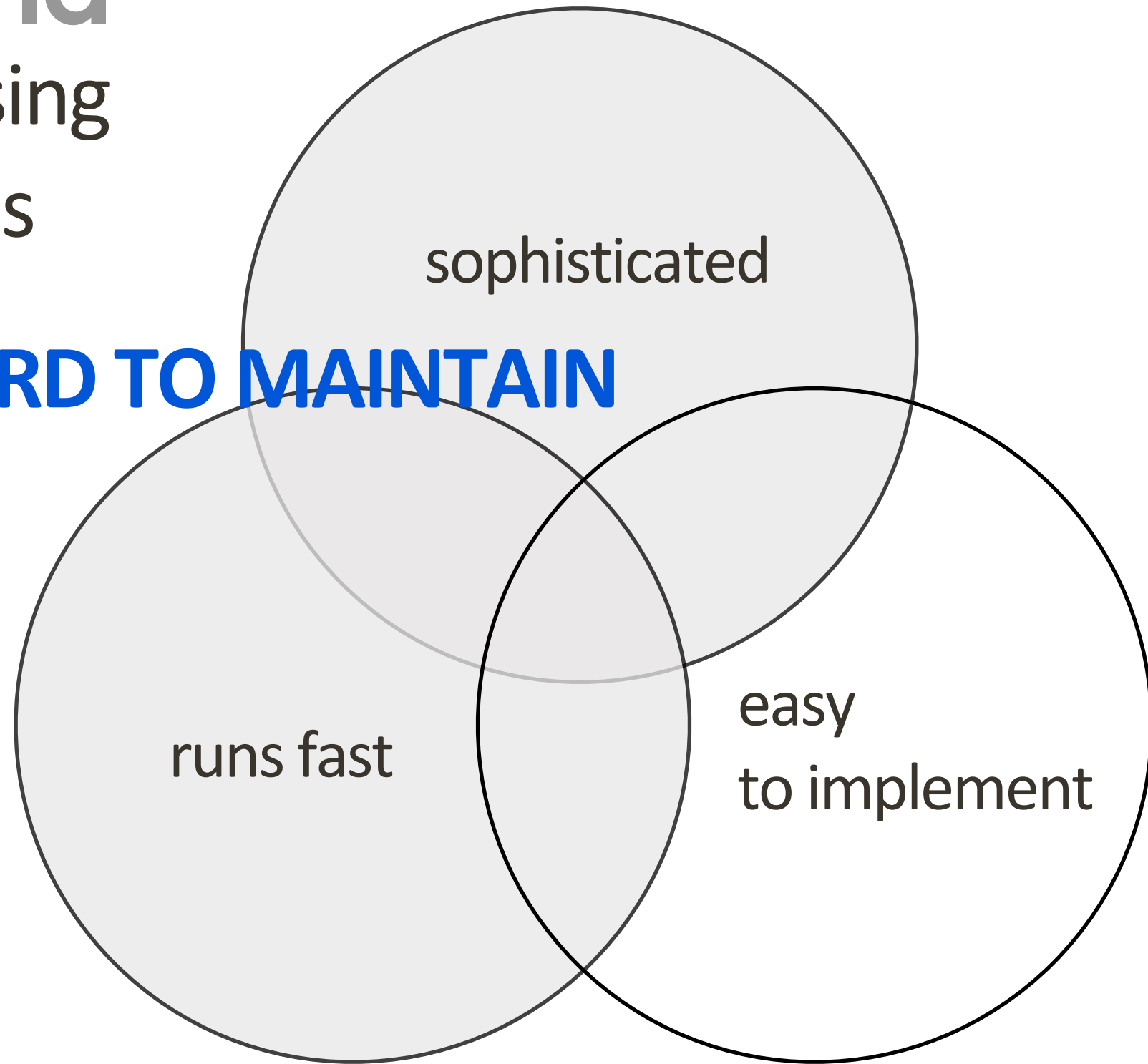
Trilemma of processing techniques



Choose 2

Trilemma of processing techniques

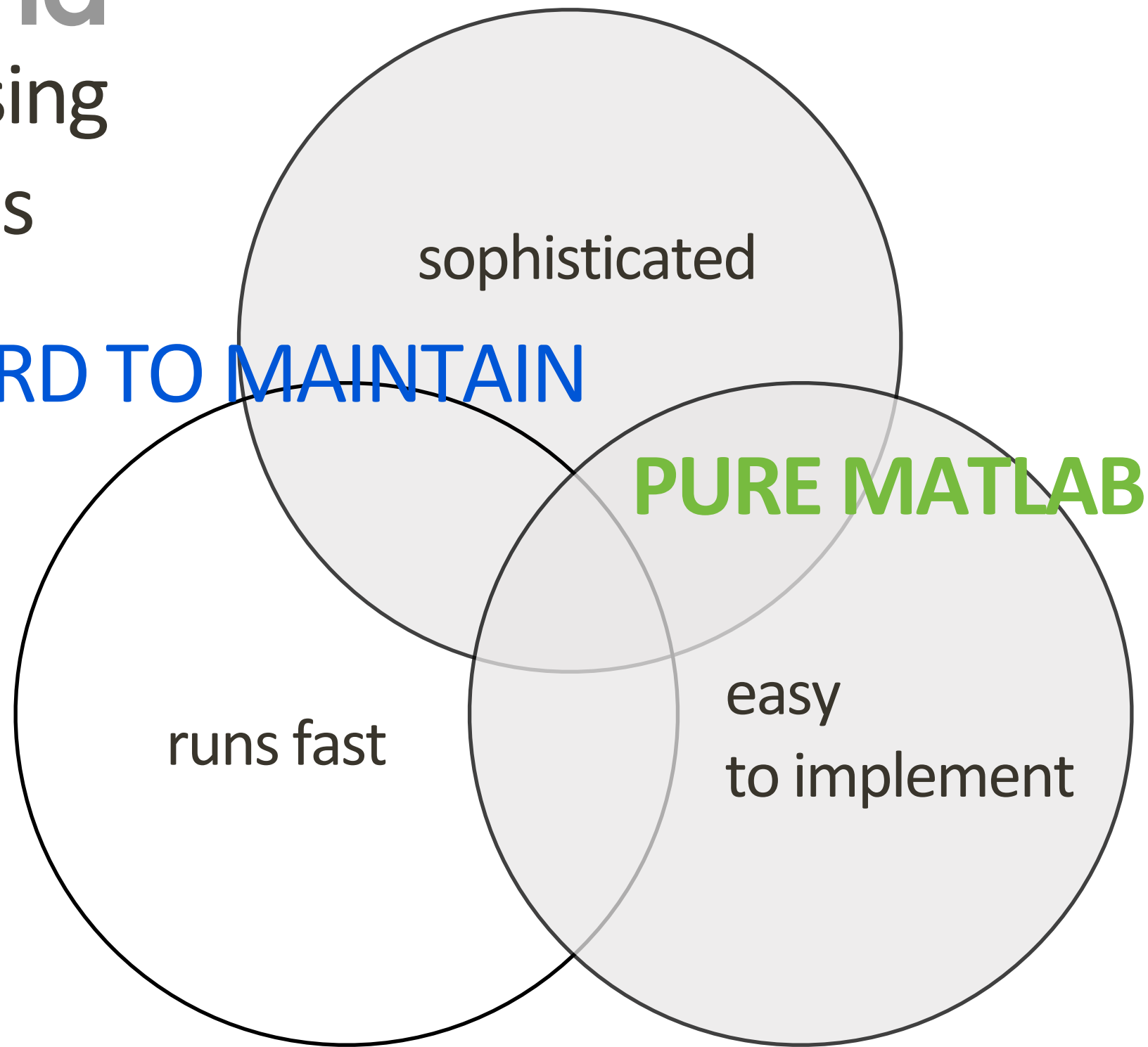
HARD TO MAINTAIN



Choose 2

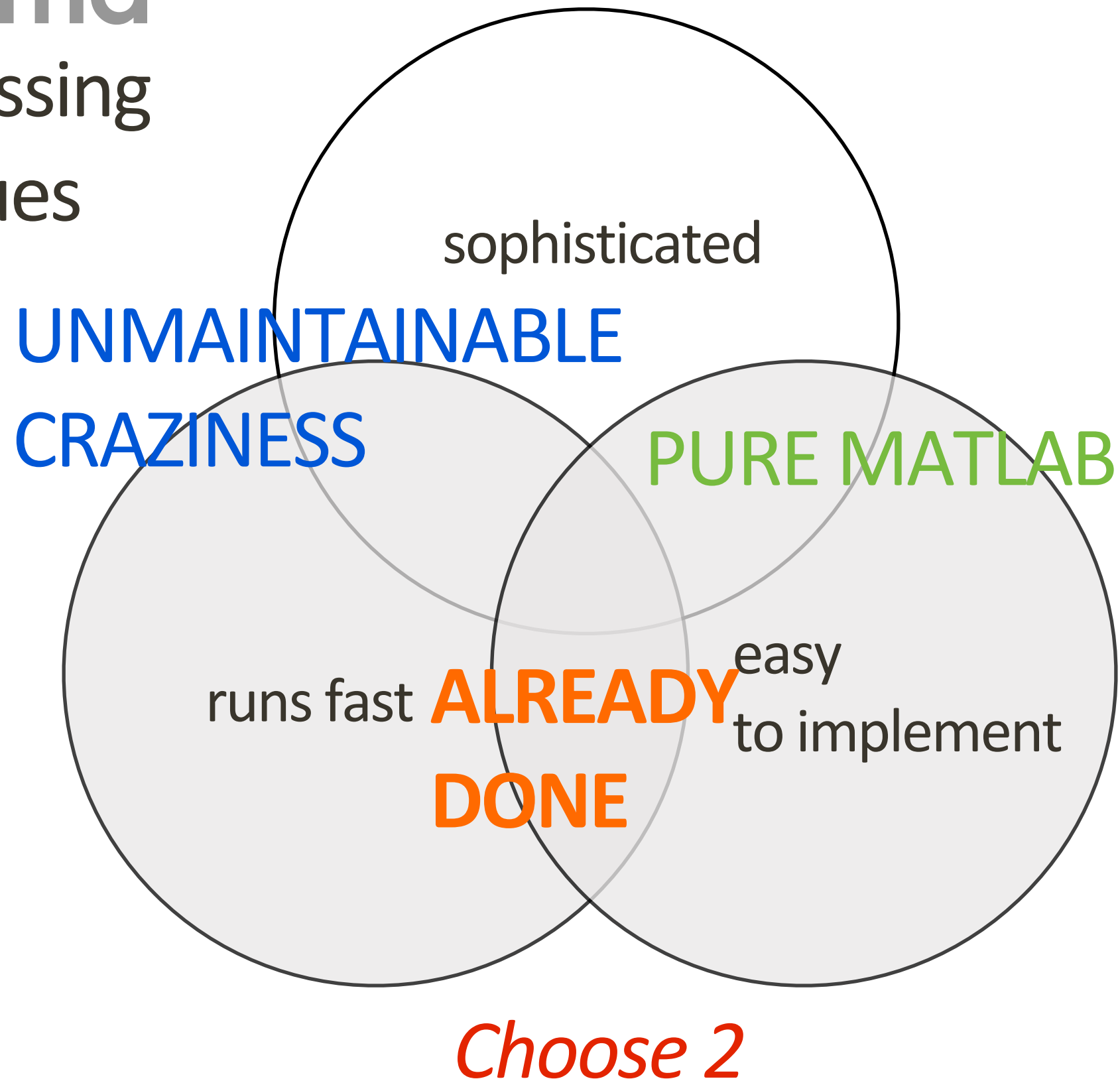
Trilemma of processing techniques

HARD TO MAINTAIN



Choose 2

Trilemma of processing techniques



Postulate

One *person* deal with **algorithm**
and *another* deal with **code?**



Projects get taken-over and inevitably stagnate...



Low-level fork



Projects get taken-over and inevitably stagnate...



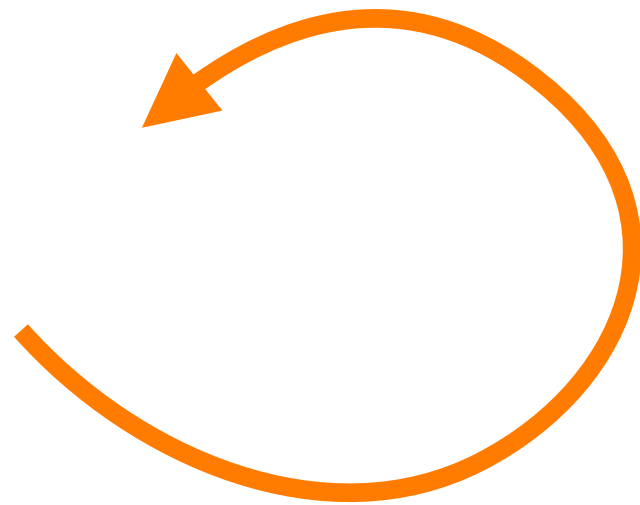
tweaks



different tweaks

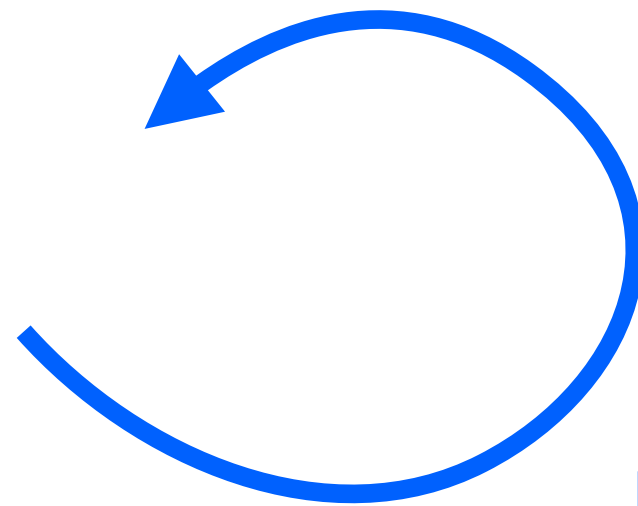


Projects get taken-over and inevitably stagnate...



feature stagnation

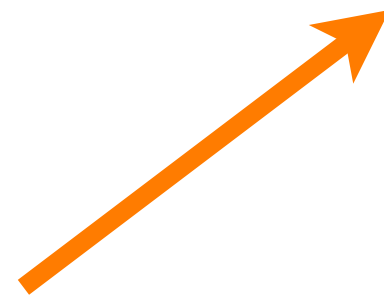
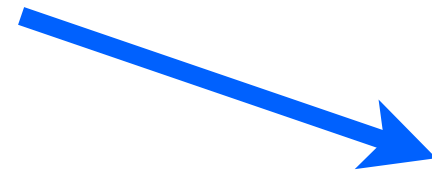
Projects get taken-over and inevitably stagnate...



low-level fork
unmaintainable



Basically there's only one clear way around this



Which Language?

*Why use the same
language?*

algorithm design

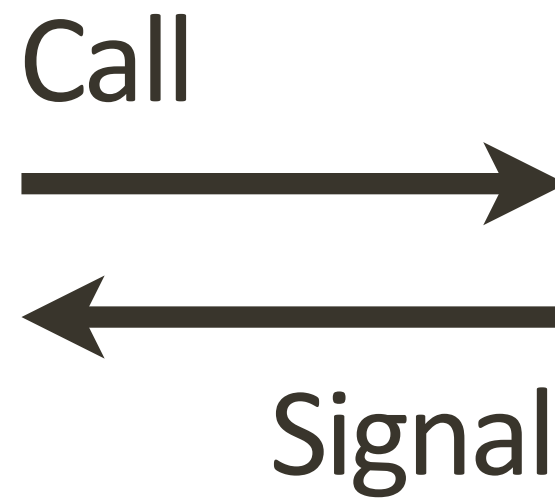
- flexible
- easy to read
- easy to debug
- reflects math
- encourage experiments

computation engine

- well-defined behaviour
- mature compiler
- low-level access
- allows tweaking
- parallel systems

**algorithm
design**

**computation
engine**



**algorithm
design**

**computation
engine**

shell scripts

Call



Signal

seismic utilities

**algorithm
design**

**computation
engine**

**workflow
management**

Call



Signal

**computation
block**

**algorithm
design**

**computation
engine**

C/C++/F90

Call



Signal

BLAS/FFTW/MKL

**algorithm
design**

**computation
engine**

MATLAB

Call



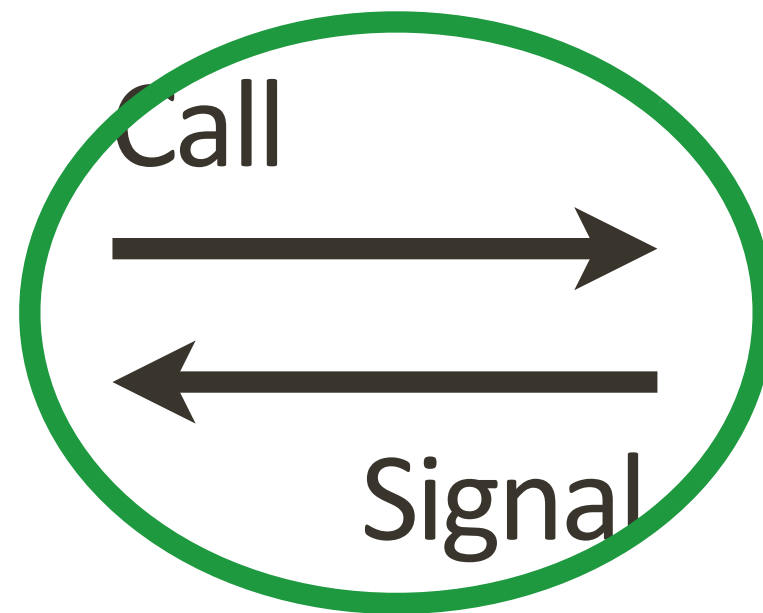
Signal

**LAPACK/
ScaLAPACK/
MEX_files**

**algorithm
design**

**computation
engine**

MATLAB

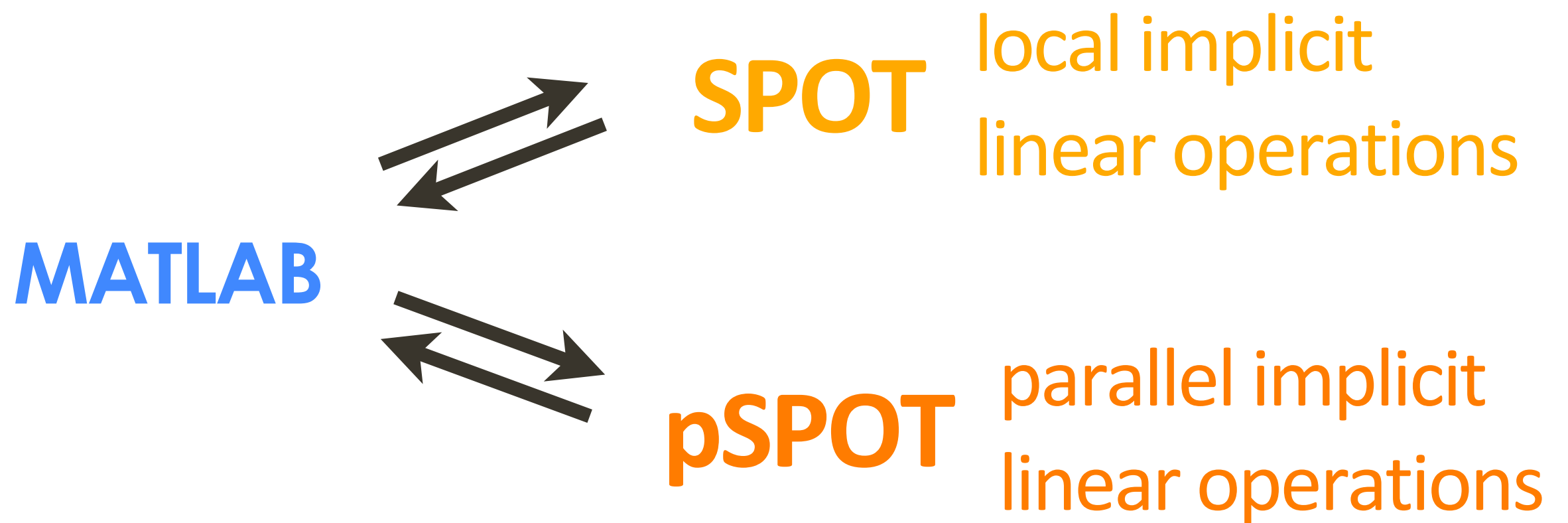


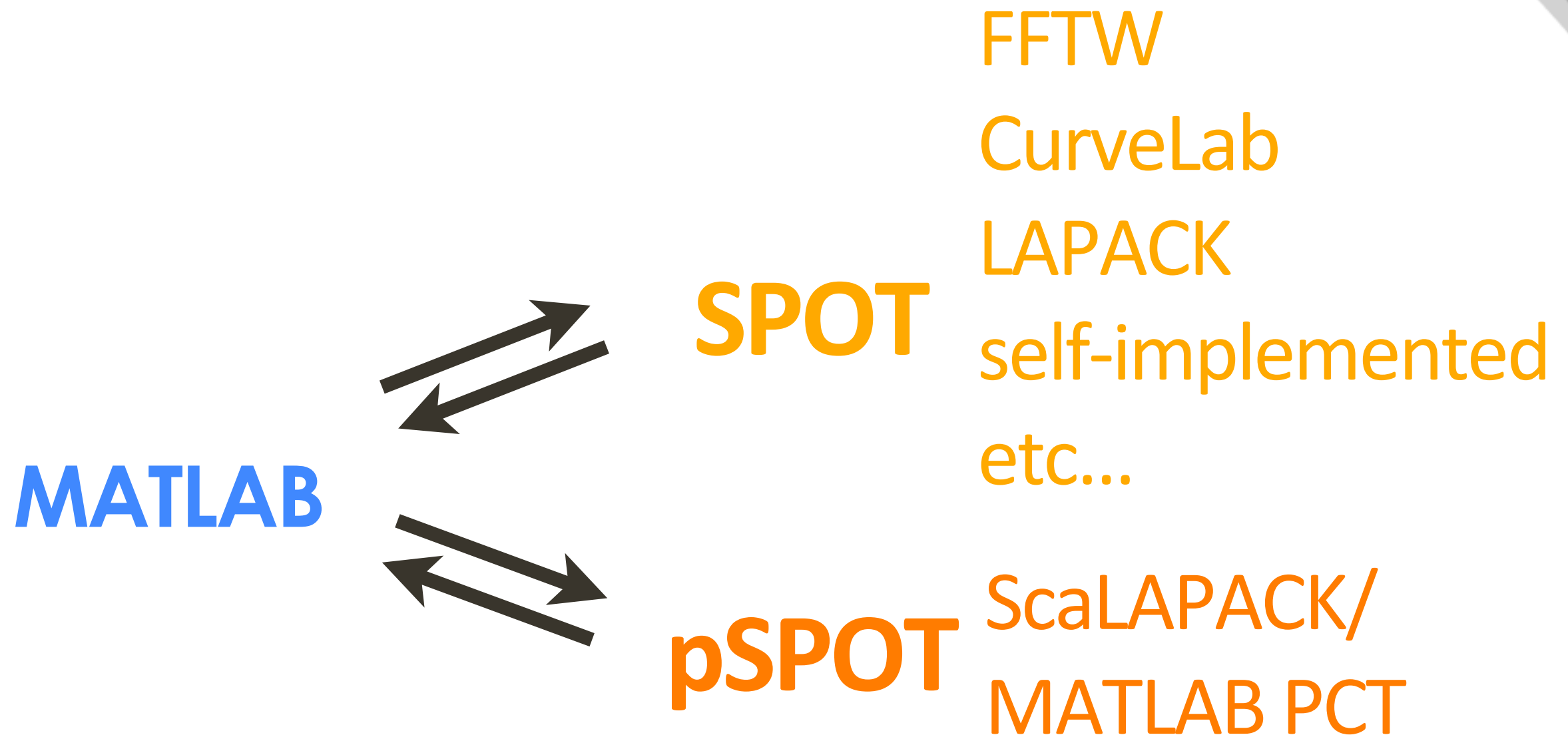
**LAPACK/
ScaLAPACK/
MEX_files**

The all-important API

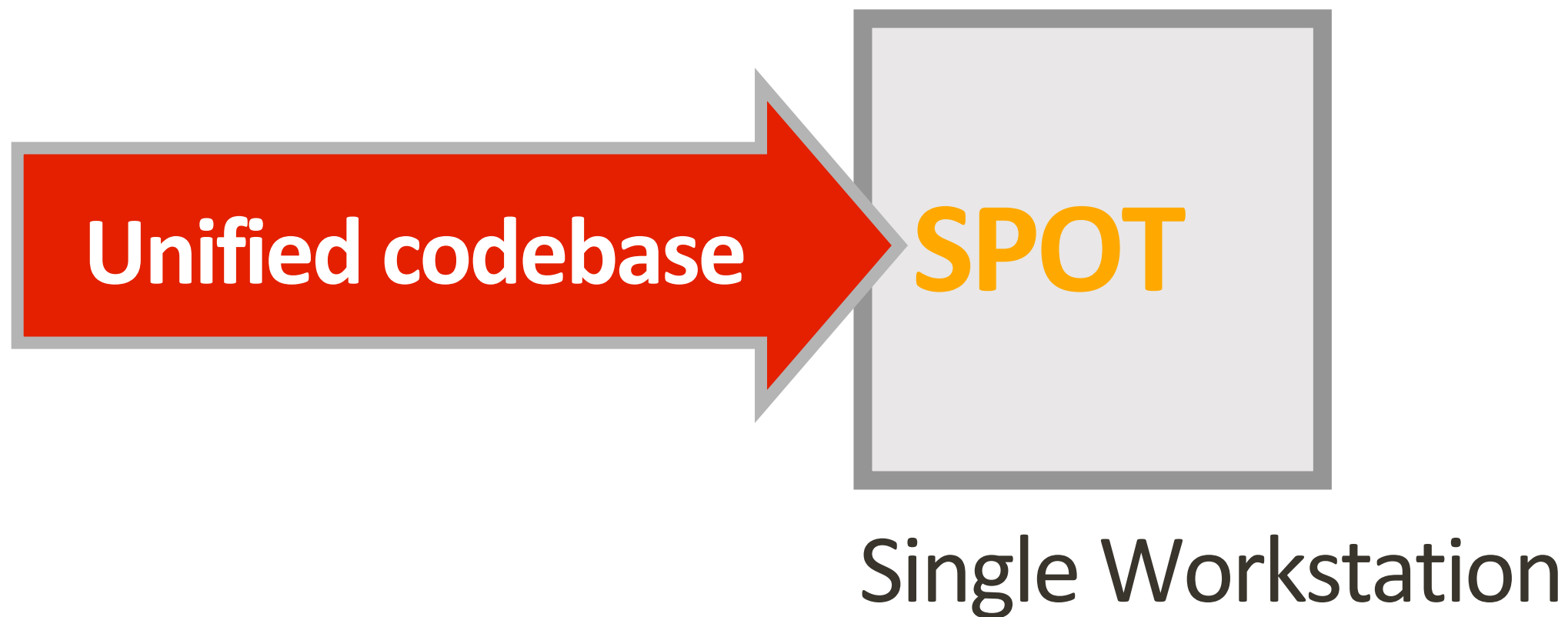
*currently, we program in MATLAB,
but our API for computation is
based on*

SPOT
pSPOT





With everything written in terms of operators, there's scalability



With everything written in terms of operators, there's scalability



Unified codebase



pSPOT

Parallel MATLAB

worker clusters for prototype

With everything written in terms of operators, there's scalability



Unified codebase

modified pSPOT
+ more API?

Industry production machine

Industry uptake

As long as the algorithms are implemented in terms of API, should allow a clear path to industry by providing the proprietary computation engine

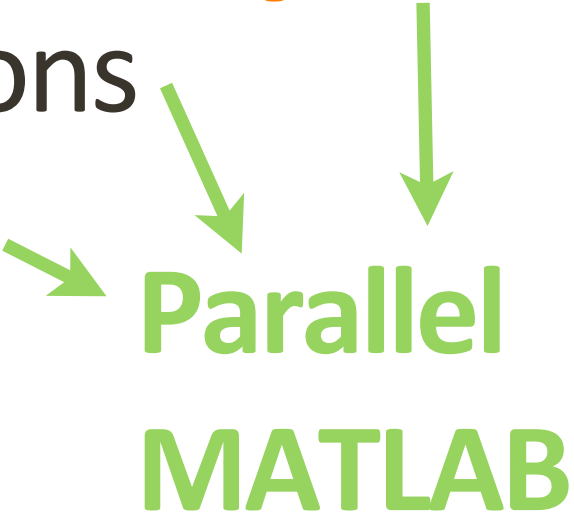
API needs to provide...

- 1) Linear operations
- 2) Element-wise operations
- 3) Reduction operations

Currently...

- 1) Linear operations **SPOT** **pSPOT**
- 2) Element-wise operations
- 3) Reduction operations

Currently...

- 1) Linear operations **SPOT** **pSPOT**
 - 2) Element-wise operations
 - 3) Reduction operations
- Parallel
MATLAB**
- 

Future

- 1) Linear operations **SPOT** **pSPOT**
 - 2) Element-wise operations
 - 3) Reduction operations
- Outside
framework?
JavaSeis?**
-

Coordinate free?

MATLAB is *good* with coordinate-free linear algebra, but *bad* at handling metainfo

Seismic tools/framework are *designed* to deal with the metainfo

Coordinate free?

Abstracting away numerical
gruntwork allows for *clean*
algorithm description & *fast*
prototyping in MATLAB

Existing cases of MATLAB/JavaSeis
working side-by-side

Collaborations

Abstracting away numerical
gruntwork facilitates
collaborations with experts in
applied math & computer science

Issue of signalling

Large data volumes should probably be only “visible in whole” to the computation framework

Need to be managed by separate **persistent** process

Issue of signalling

Choice of inter-process communication

Sockets?

Signals?

Message Passing?

Semaphores?

Signal files?

Memory-mapping?

Issue of signalling

Representation of large datasets

Need a standard “alias” for a large dataset that can only be **probed** by the **algorithm** (*e.g. norms, averages, etc*); All other **computation** needs to be done via **external framework**

Issue of signalling

Representation of large datasets

Current & ongoing work at SLIM

Need a standard “alias” for a large dataset that can only be **probed** by the **algorithm** (*e.g. norms, averages, etc*); All other **computation** needs to be done via **external framework**

Promotes better students

Scheme will fail without **adapting good coding practices** and separation of critical/noncritical code sections

Promotes better students

Encourages **experimentation** of **interesting ideas** using high-level language

Promotes better students

Encourages **collaboration of code**
by nature of having more **readable**
codebase

Goal

Practical expression of
sophisticated ideas