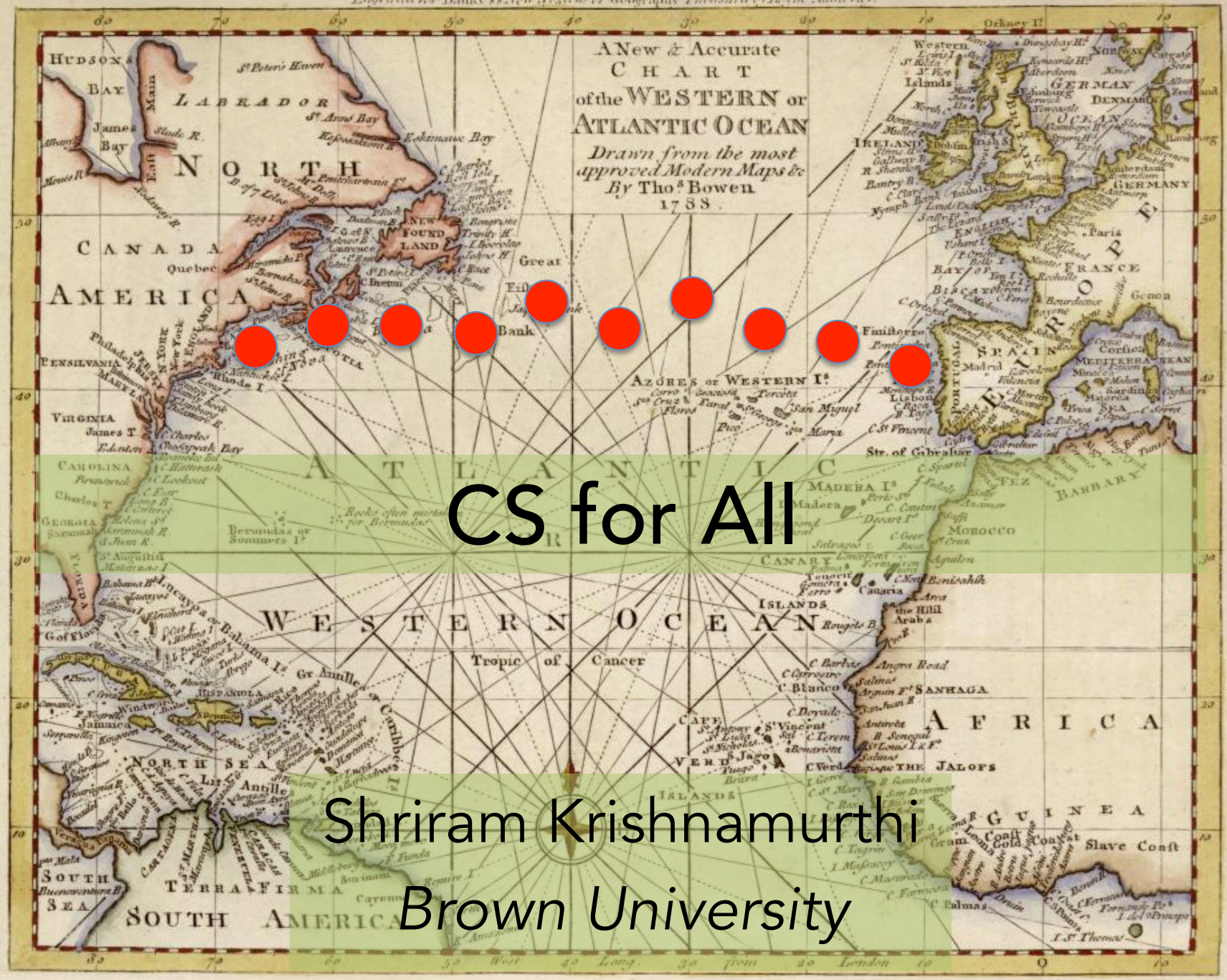
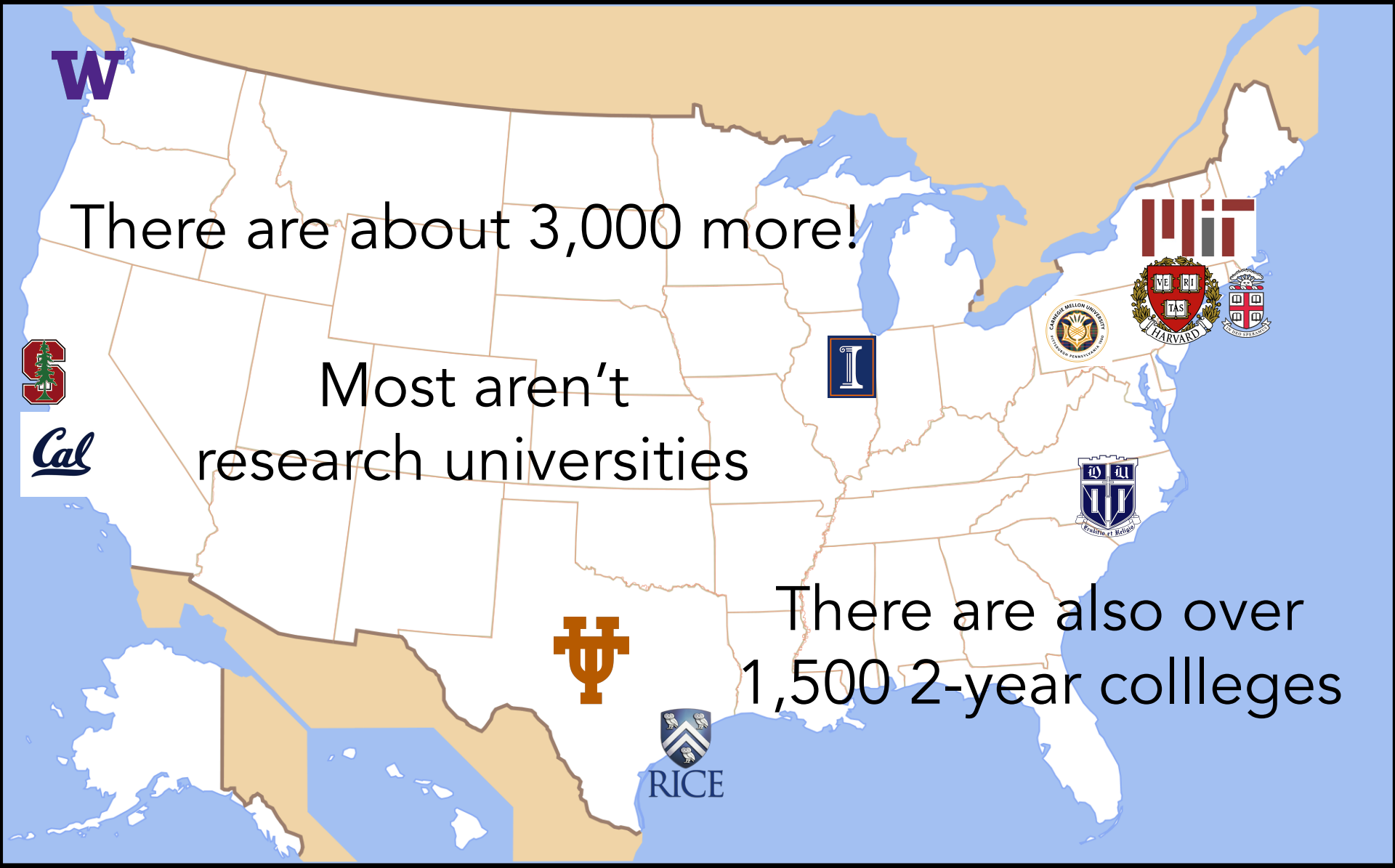
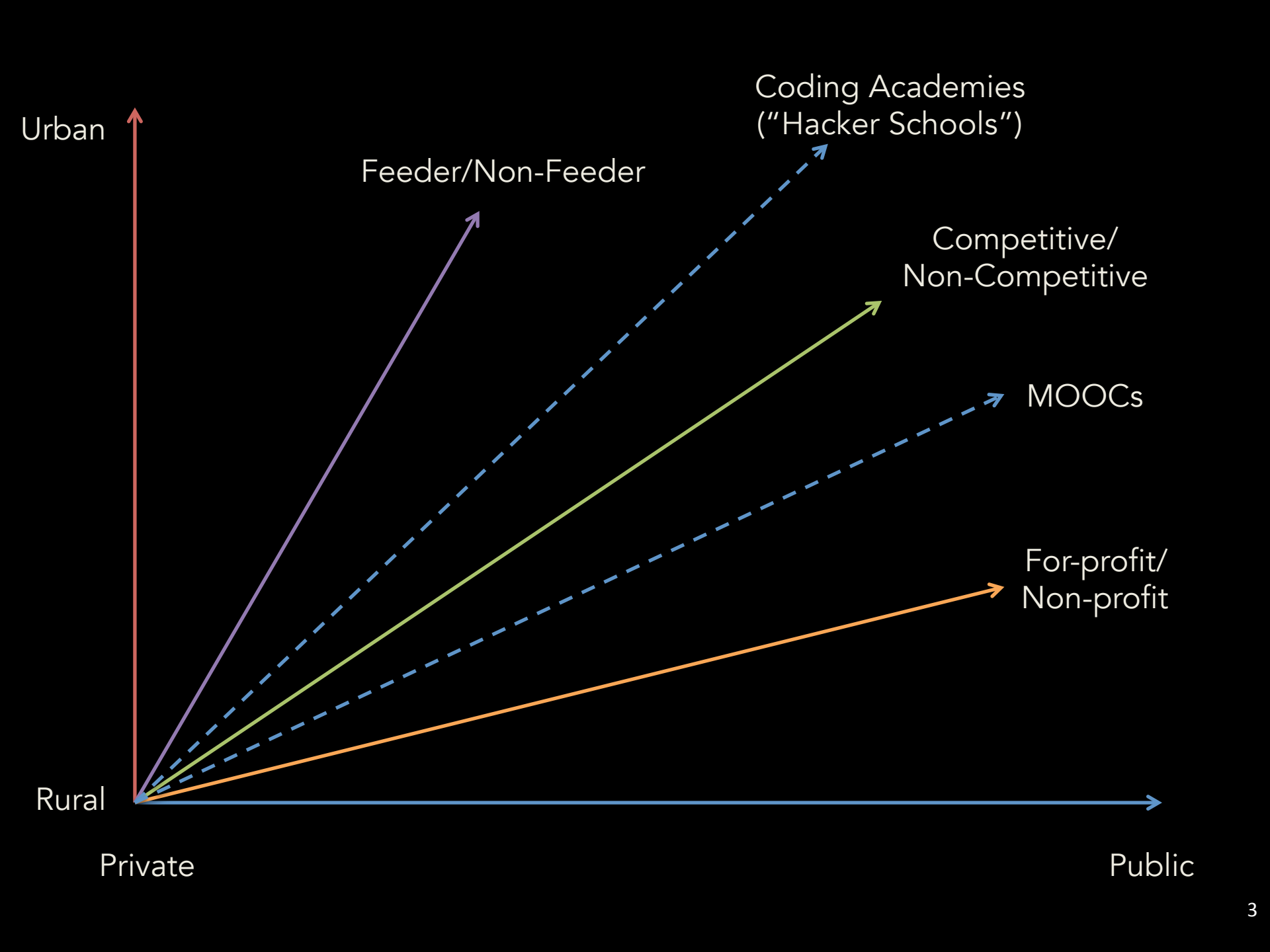


CS for All

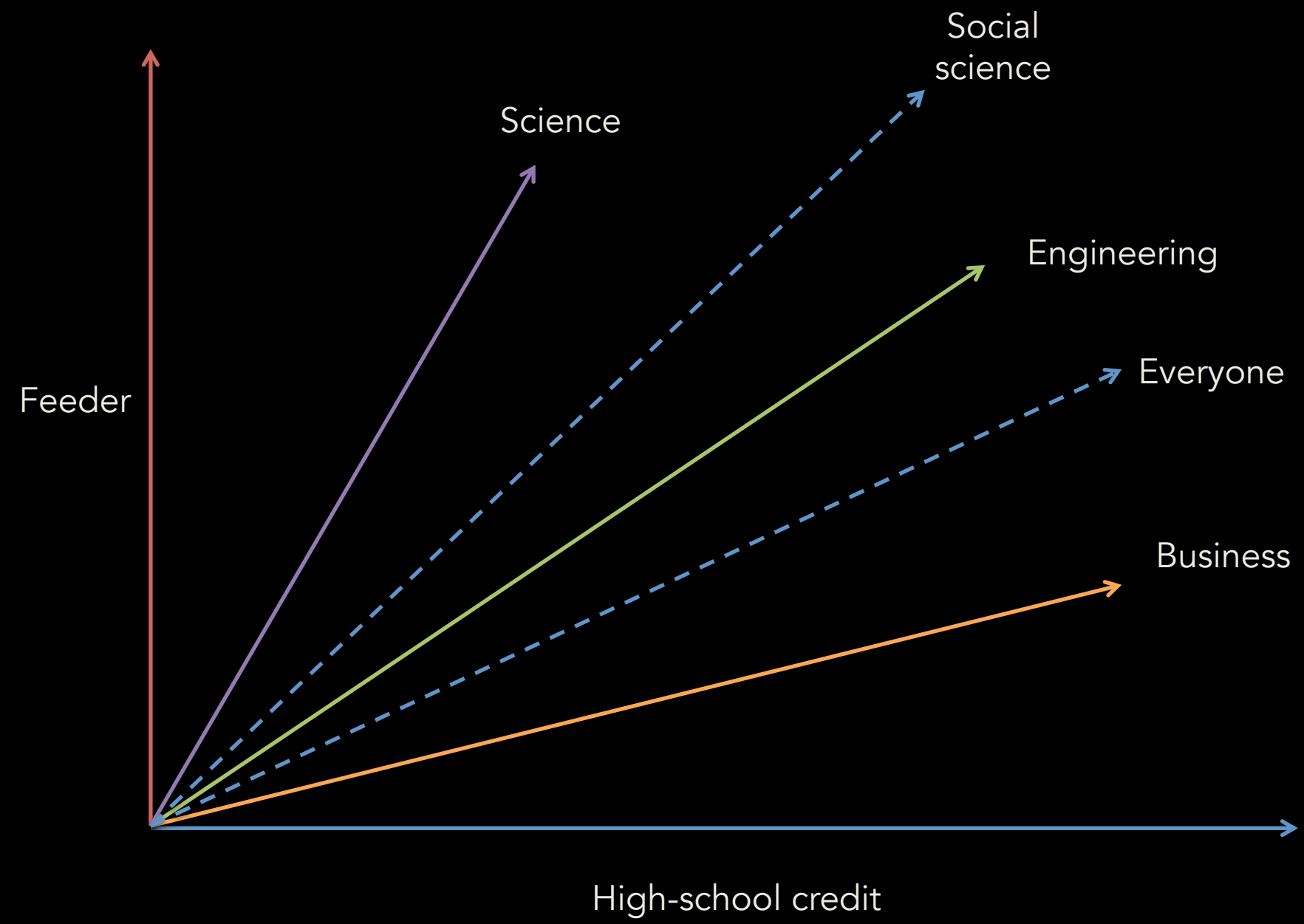
Shriram Krishnamurthi

Brown University

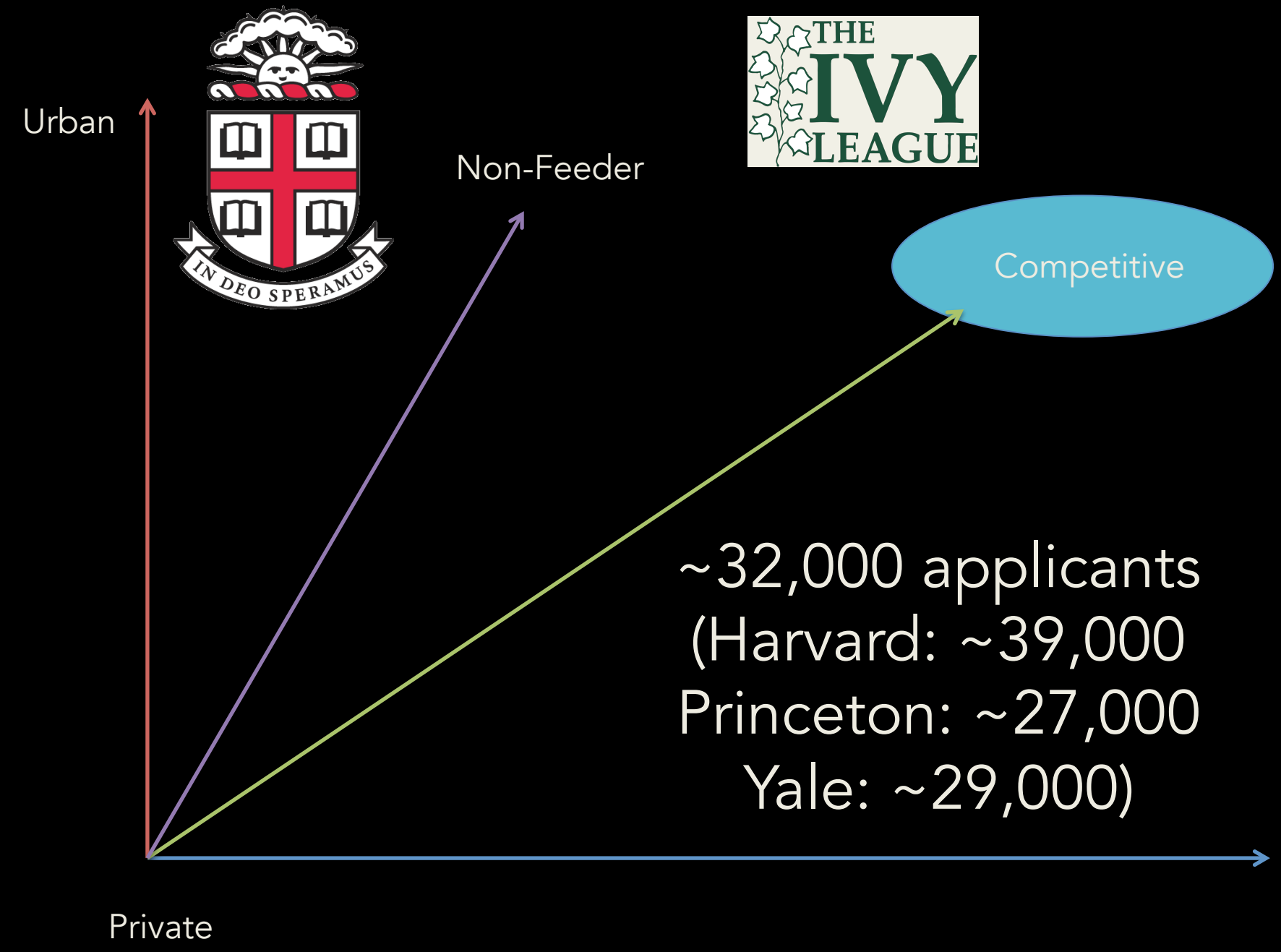

There are about 3,000 more!
Most aren't
research universities
There are also over
1,500 2-year collleges
RICE


Urban
Coding Academies
("Hacker Schools")
Feeder/Non-Feeder
Competitive/
Non-Competitive
MOOCs
For-profit/
Non-profit
Rural
Private
Public

Design Forces


Social
science
Science
Engineering
Everyone
Business
Feeder
High-school credit

Computing for All


Urban
Non-Feeder
Competitive
Private
~32,000 applicants
(Harvard: ~39,000
Princeton: ~27,000
Yale: ~29,000)

Brown CS

CS is the #1 major at Brown

25% bigger than next biggest major

Approximately 12% of university

Without sacrificing rigor!

About 40% to Google; MS, Fb, …

What About the Rest?

Several strategies for rest of campus

Easy way: Make it (meet) a requirement

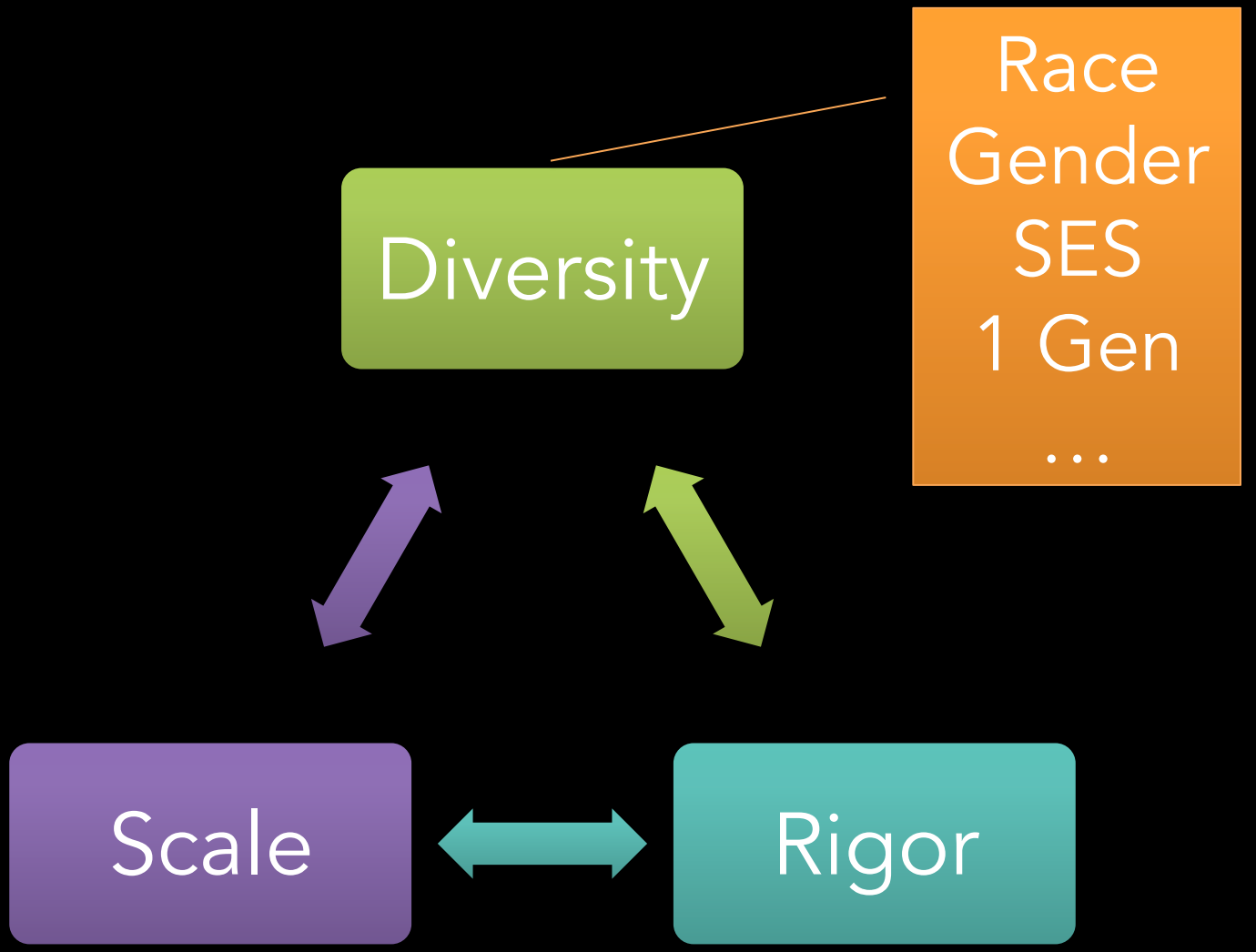
Hard way: Everything else!

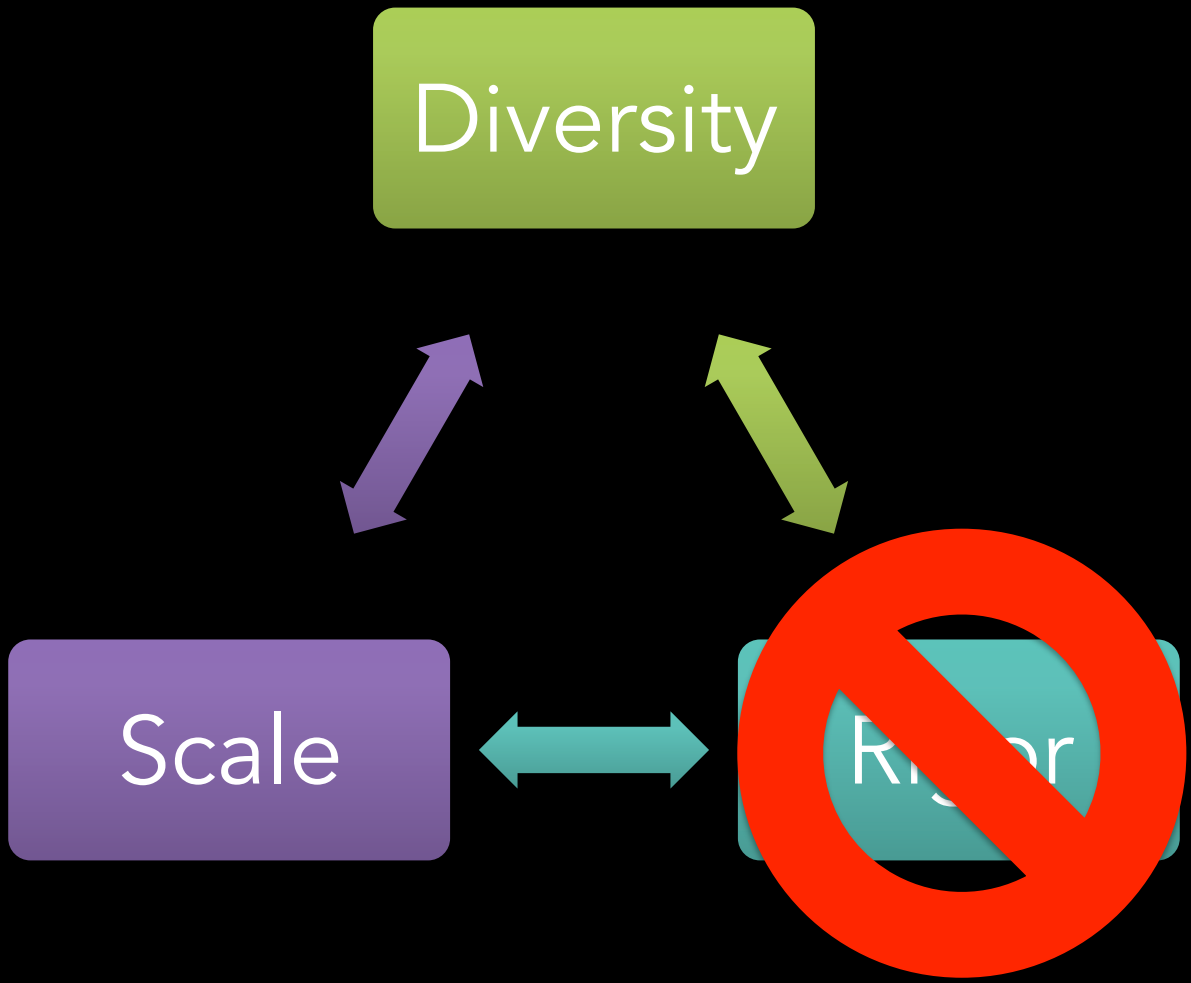
Bootstrap: Computational Modeling
in Algebra, Physics, and Data Science
for all students

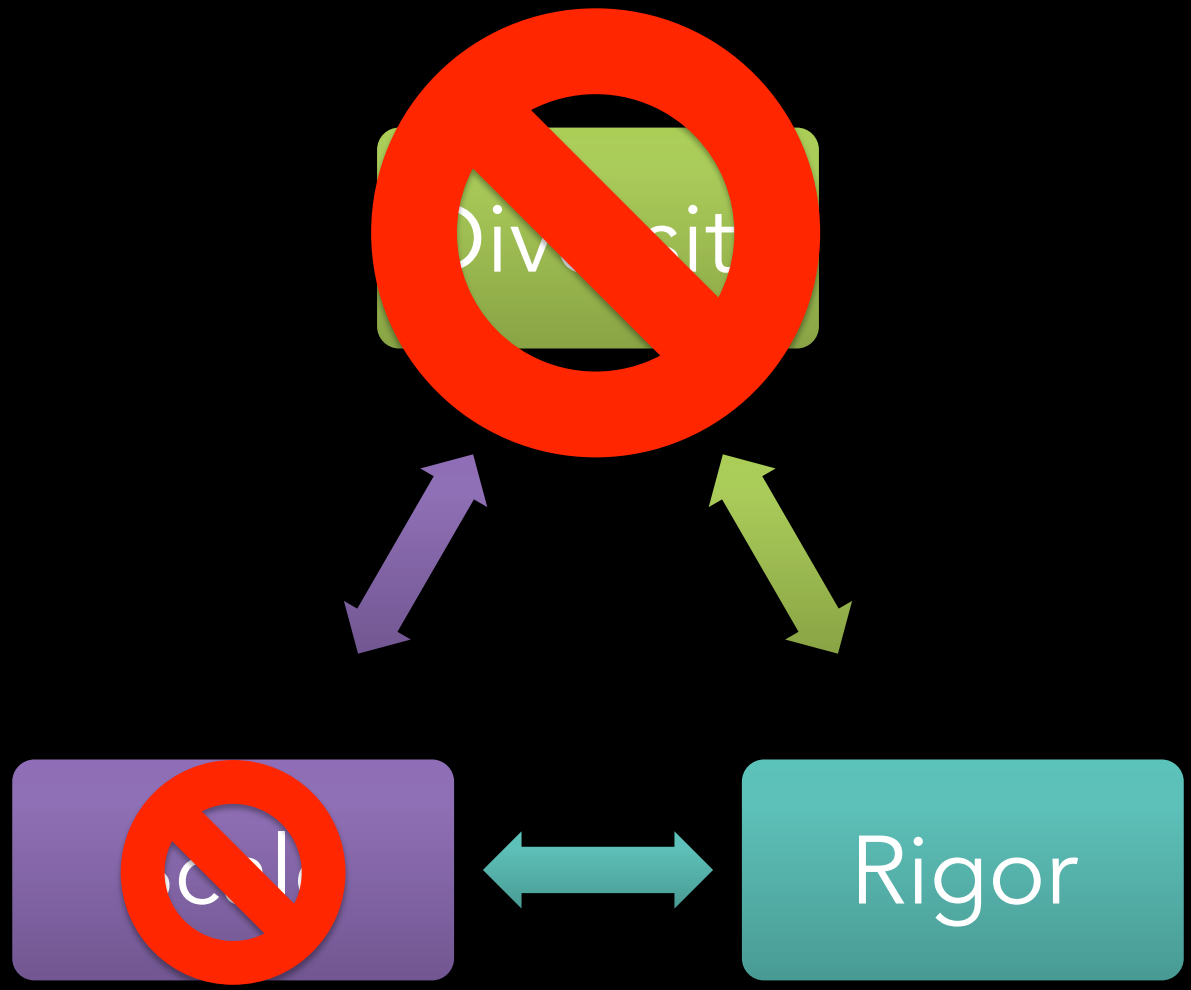
One of the largest CS outreach programs
Part of White House's CS4All program

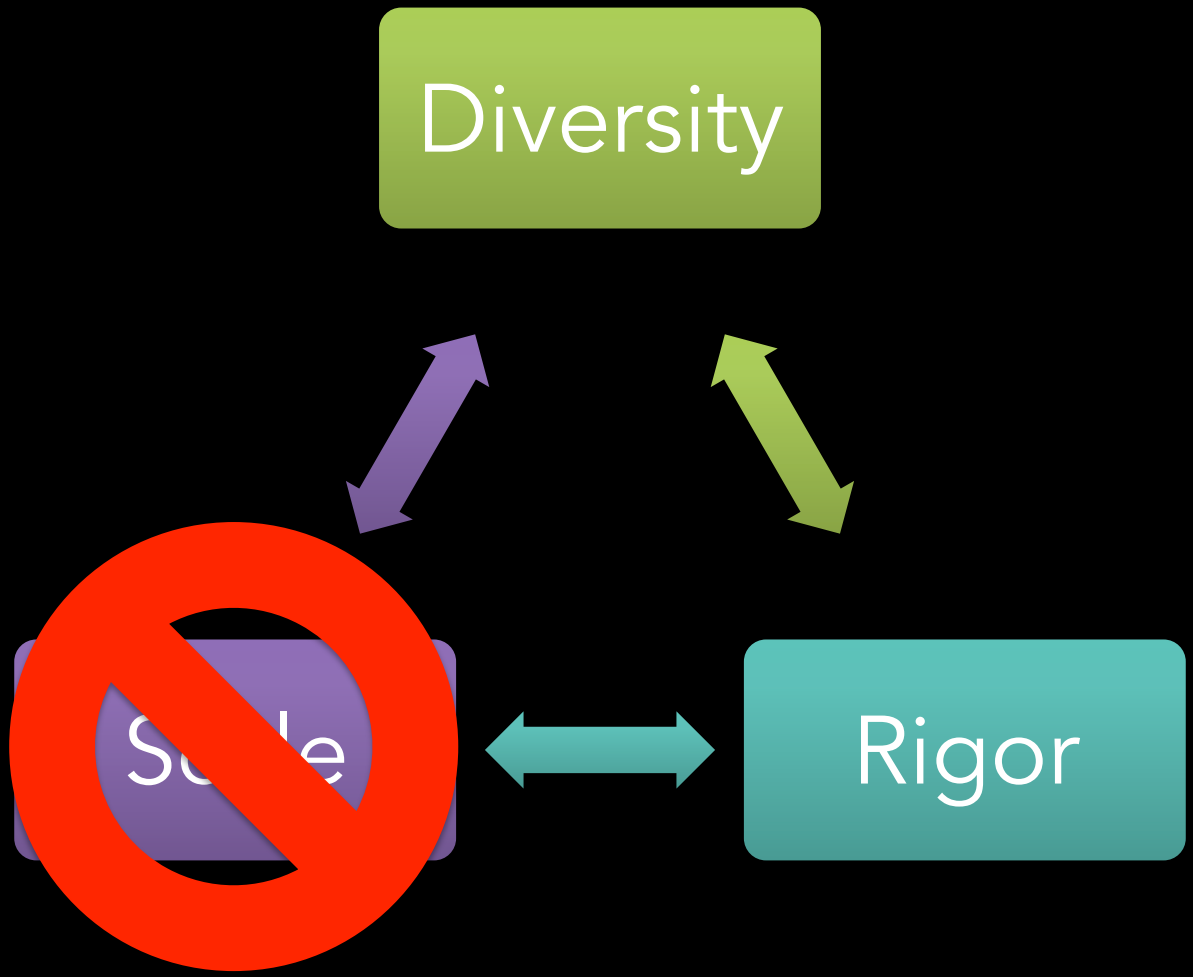
Curriculum Design is an Engineering Problem

What are your design constraints?


Diversity
Race
Gender
SES
1 Gen
…
Scale
Rigor


Diversity
Scale


Rigor

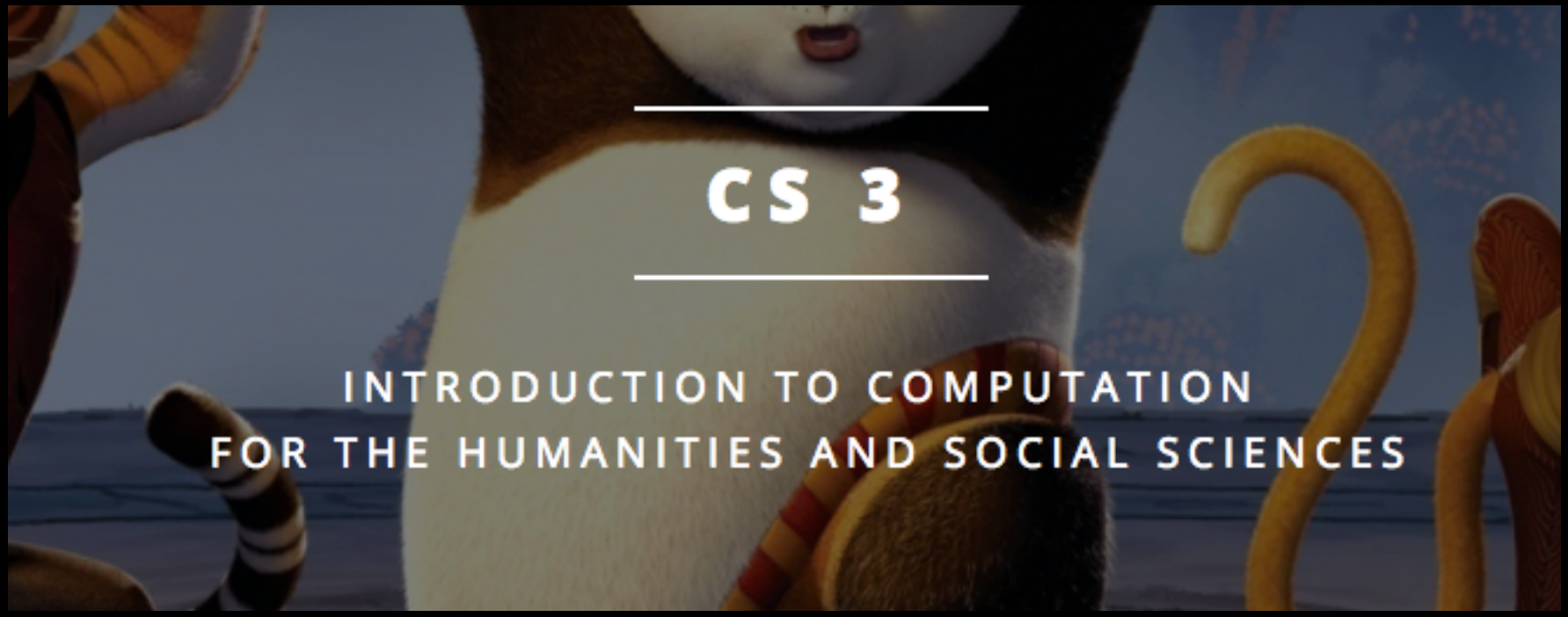

Diversity
Rigor

Brown CS

CSCI0020	(CS002)	**The Digital World**
CSCI0030	(CSCI0931)	**Introduction to Computation for the Humanities and Social Sciences**
CSCI0040	(CS004)	**Introduction to Scientific Computing and Problem Solving**
CSCI0050		**A Data-Centric Introduction to Programming**
CSCI0080		**A First Byte of Computer Science**
CSCI0090-A	(CS009-3)	Building a Web Application
CSCI0090-B	(CS009-1)	Computers and Human Values
CSCI0090-C	(CS009-2)	Talking with Computers
CSCI0100		**Data Fluency for All**

Bootstrap

Incorporate into required school courses
(Algebra, Social Studies, Science)
with measured transfer



Three month-long projects
Problems taken from target subjects
A month of Excel!
Final output is a report, not program

SIGCSE 2018 Tutorial

Saturday 2-5pm

From Spreadsheets to Programs:
Reconciling Data Science and CS1

Politz, Fisler, Krishnamurthi, Lerner

The Immaturity of CS Ed

Where is the science for curriculum engineering?

Diction

Our diction is still stuck with languages

"We teach Java"
"We teach Python"

Not always necessary; certainly not sufficient

Principles vs. Platforms

A computing platform (Arduino, drone, …):

- represents itself
- represents something bigger than itself

Failing to articulate learning objectives means we conflate them (and skip the latter)

Continuity

Later classes don't pick up on earlier ones

- How many of your faculty really know what is taught in the intro class?
- How many care?

Early faculty don't want to know what is in later classes

- "Let me teach Haskell and leave me alone!"

New Challenges

Where are

- embedded computing
- distributed computing
- data science?

Each has fundamentally new requirements

Can't just keep doing `for` loops (or objects)

Pressure from Below

Students increasingly come to college with quite sophisticated backgrounds

Need to remove them from the general student pool

Worse, words ≠ knowledge

Pushing Downward

What is your CS-in-schools initiative?

Do you treat it as more than a hobby?

What are its design criteria?
(Diversity, rigor, scale?)

Plagiarism

A problem from a certification perspective

It's really a mechanism design problem

Our successful approach so far:
Peer review

Running in Place

Enrollment challenges means

- no time to innovate
- no need to attract new students
- resources are spread thin
- student quality is variable
- difficult to maintain authenticity

yet the opportunities are greater than ever