Lecture 1 — Introduction

Stanford CS343D (Winter 2023)
Fred Kjolstad

Course staff

Fred Kjolstad

AJ Root

Administria

- Syllabus at https://cs343d.github.io
- Discussion will happen through Ed in Canvas
- Office Hours
    - Fred: Monday 10–11am in Gates 486
    - AJ: Thursday 2-3pm in Gates 4A common area

Goals of the Course

- Introduce you to domain-specific and collection-oriented programming languages from the past

- Introduce you to compiler techniques to get good performance for dense and sparse applications

- Bring you to one of the frontiers of PL and compiler research

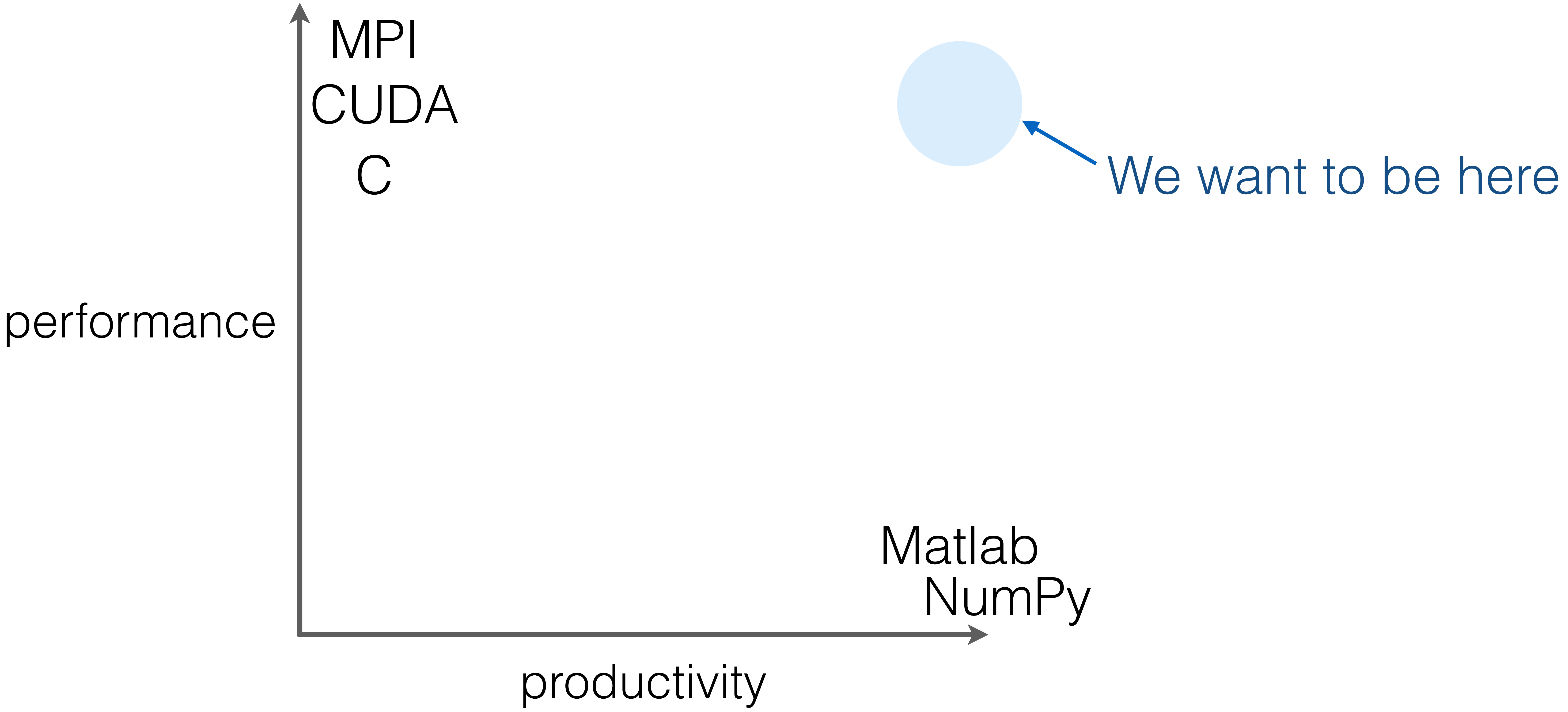
- Get you thinking about abstractions and semantics

- “What are the three biggest ideas in computer science? Abstraction, abstraction, abstraction.”
  -Paul Hudak

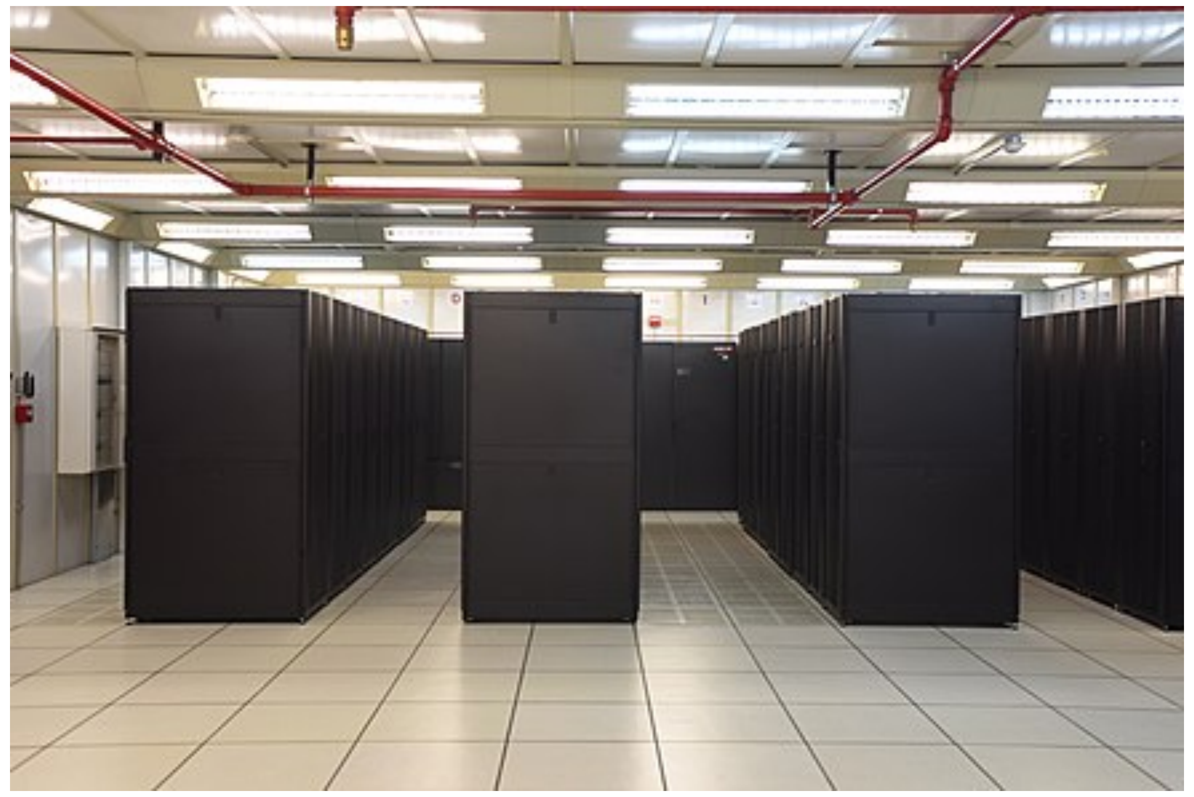
Expectations

- Read papers and engage in class (25%)
  - ~2 readings per class
  - Classes will have a lecture followed by paper discussion
  - Everyone will get a chance to lead a discussion
- Two assignments (20%)
  - MiniAPL
  - Sparse Coiteration Code Generation
- Essay (15%)
- Project (40%)

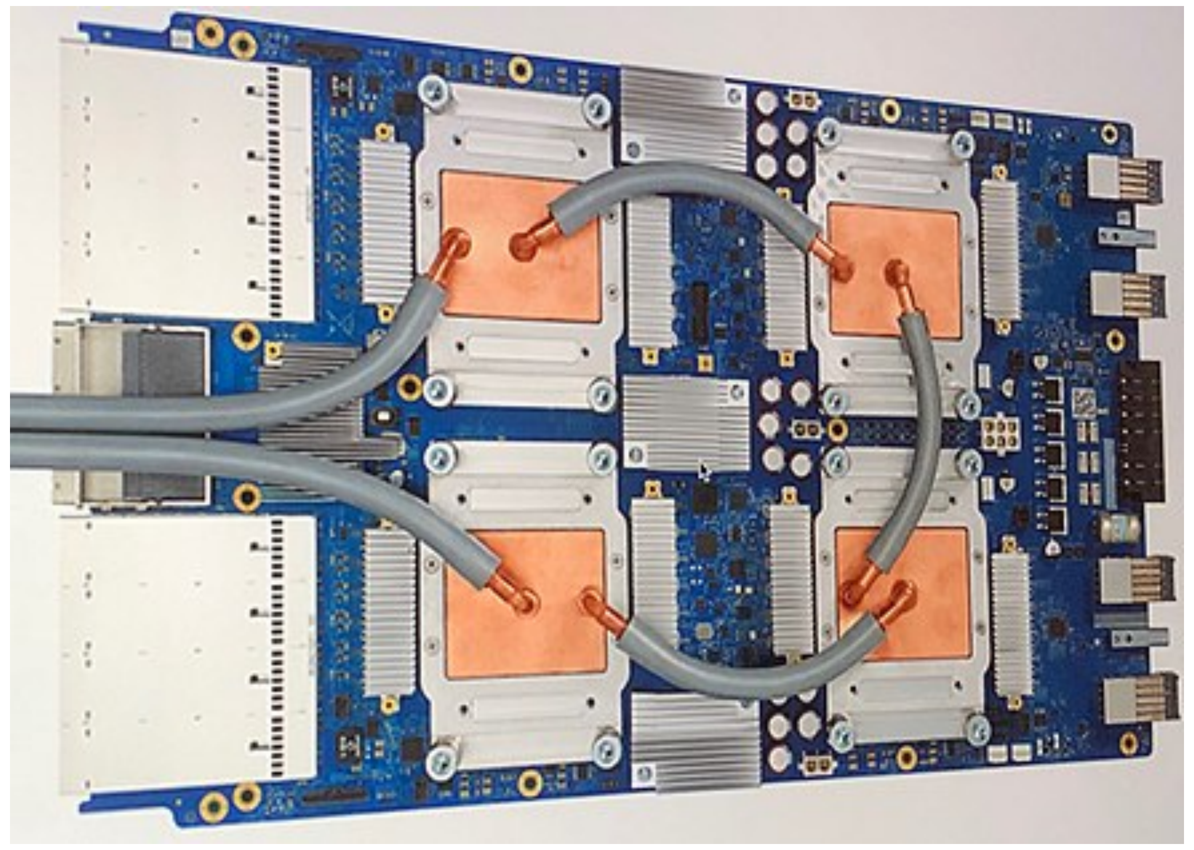
It is all about performance and productivity



Performance translates to less time and less energy



Data centers



Tensor Processing Unit

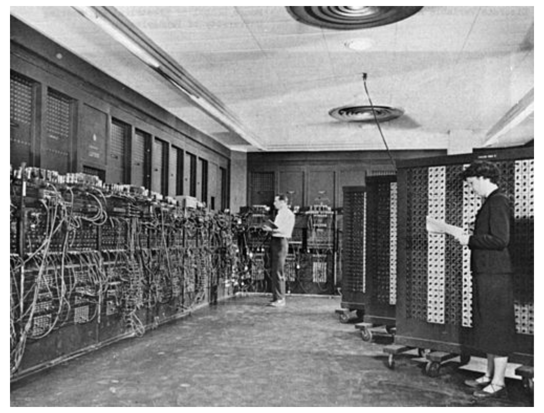
Supercomputers

Self-driving cars

Cell-phone batteries

Eras of Computing

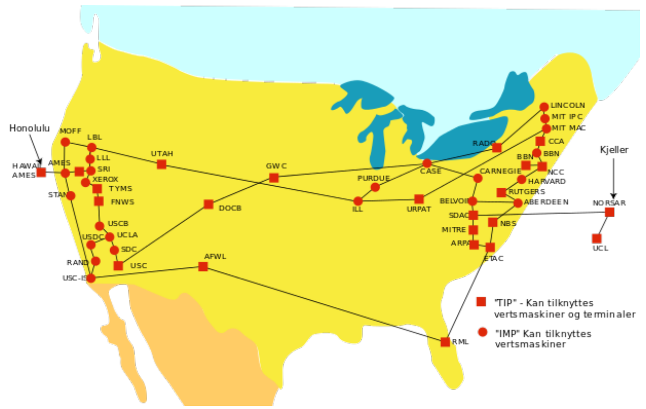
Era of simulation (1945–1965)



Era of data processing (1965–1984)

Era of Personal Computing (1984–1995)

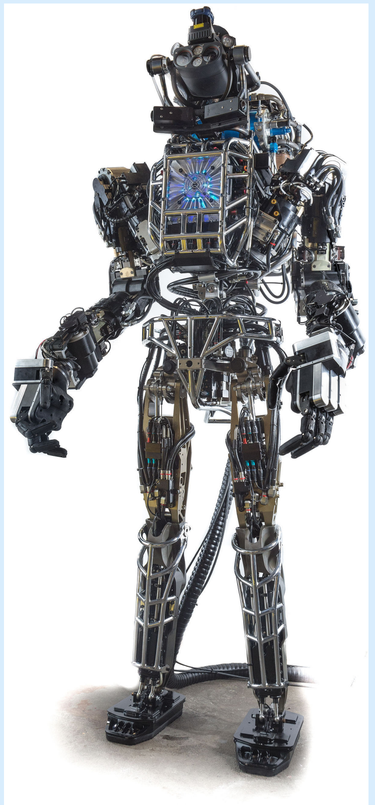
Era of communication (1995–2018)



Era of interaction (2018–????)

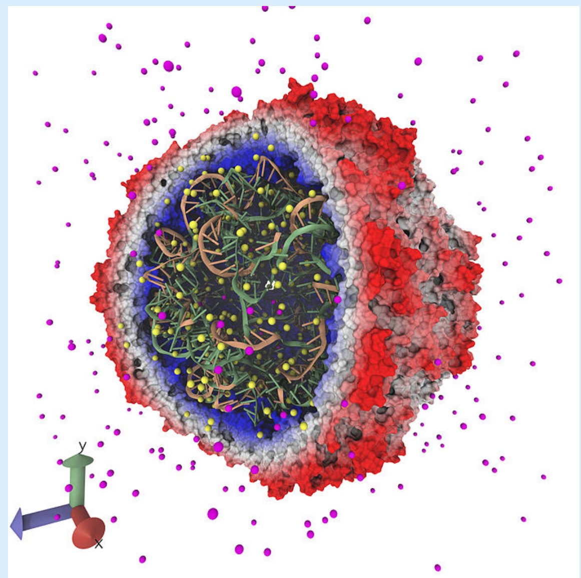
Modern applications are performance hungry

Simulation and Optimization



Robotics

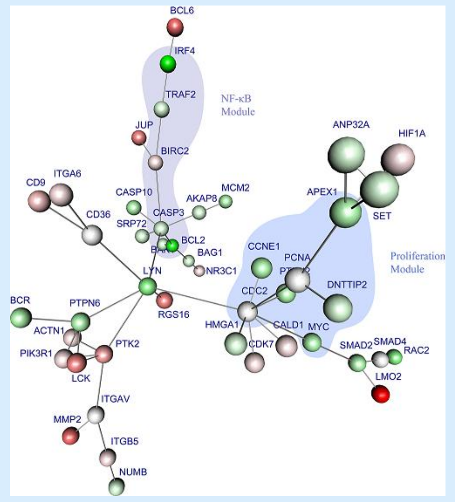
Graphics Simulations



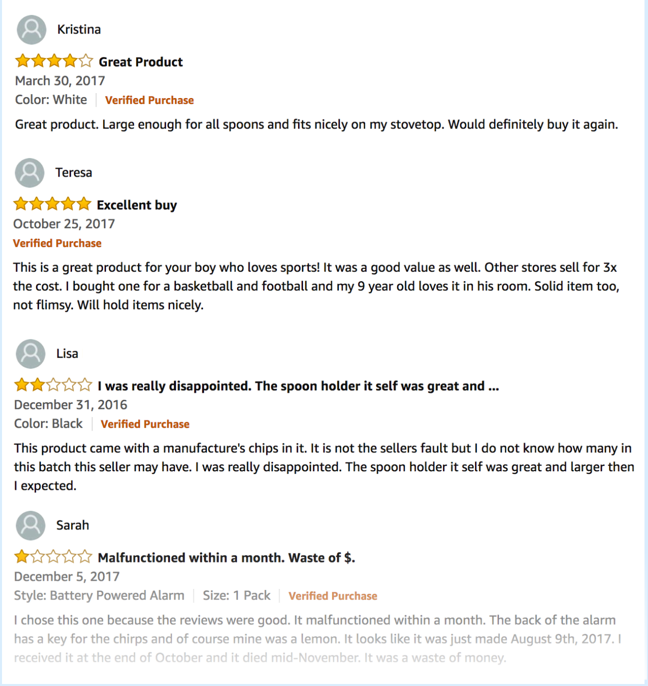
Virus Modelling

Data Analytics

Social Networks



Computational Biology



Kristina

★★★★☆ **Great Product**

March 30, 2017

Color: White | Verified Purchase

Great product. Large enough for all spoons and fits nicely on my stovetop. Would definitely buy it again.

Teresa

★★★★★ **Excellent buy**

October 25, 2017

Verified Purchase

This is a great product for your boy who loves sports! It was a good value as well. Other stores sell for 3x the cost. I bought one for a basketball and football and my 9 year old loves it in his room. Solid item too, not flimsy. Will hold items nicely.

Lisa

★★☆☆☆ **I was really disappointed. The spoon holder it self was great and ...**

December 31, 2016

Color: Black | Verified Purchase

This product came with a manufacture's chips in it. It is not the sellers fault but I do not know how many in this batch this seller may have. I was really disappointed. The spoon holder it self was great and larger then I expected.

Sarah

★☆☆☆☆ **Malfunctioned within a month. Waste of $.**

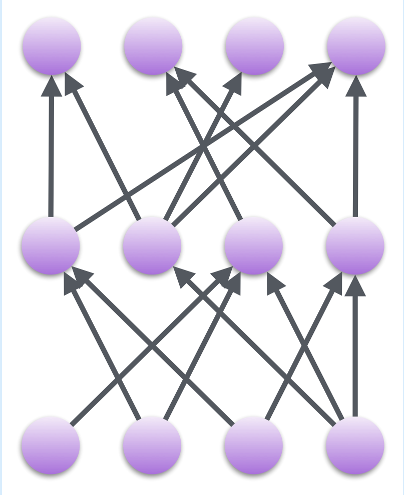
December 5, 2017

Style: Battery Powered Alarm | Size: 1 Pack | Verified Purchase

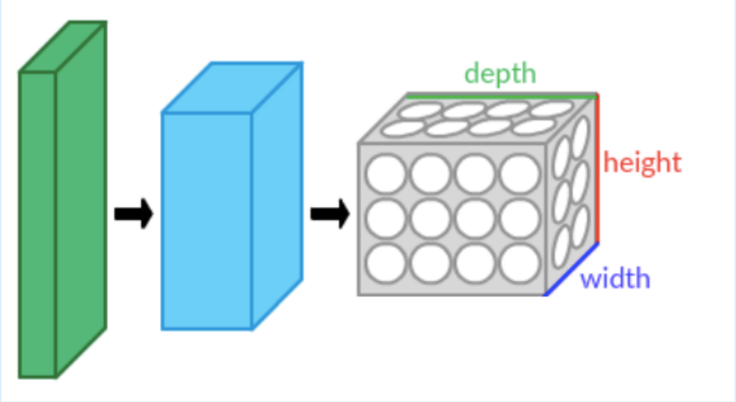
I chose this one because the reviews were good. It malfunctioned within a month. The back of the alarm has a key for the chirps and of course mine was a lemon. It looks like it was just made August 9th, 2017. I received it at the end of October and it died mid-November. It was a waste of money.

Recommender Systems

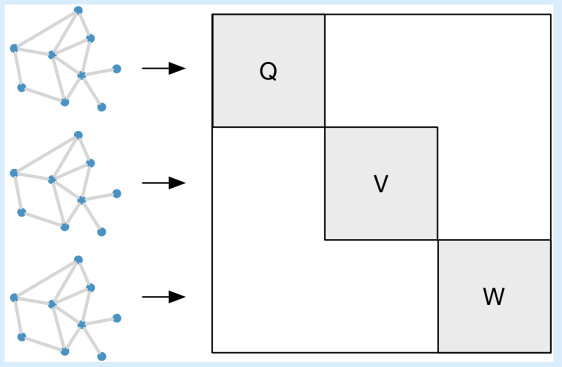
Machine Learning



Neural Networks

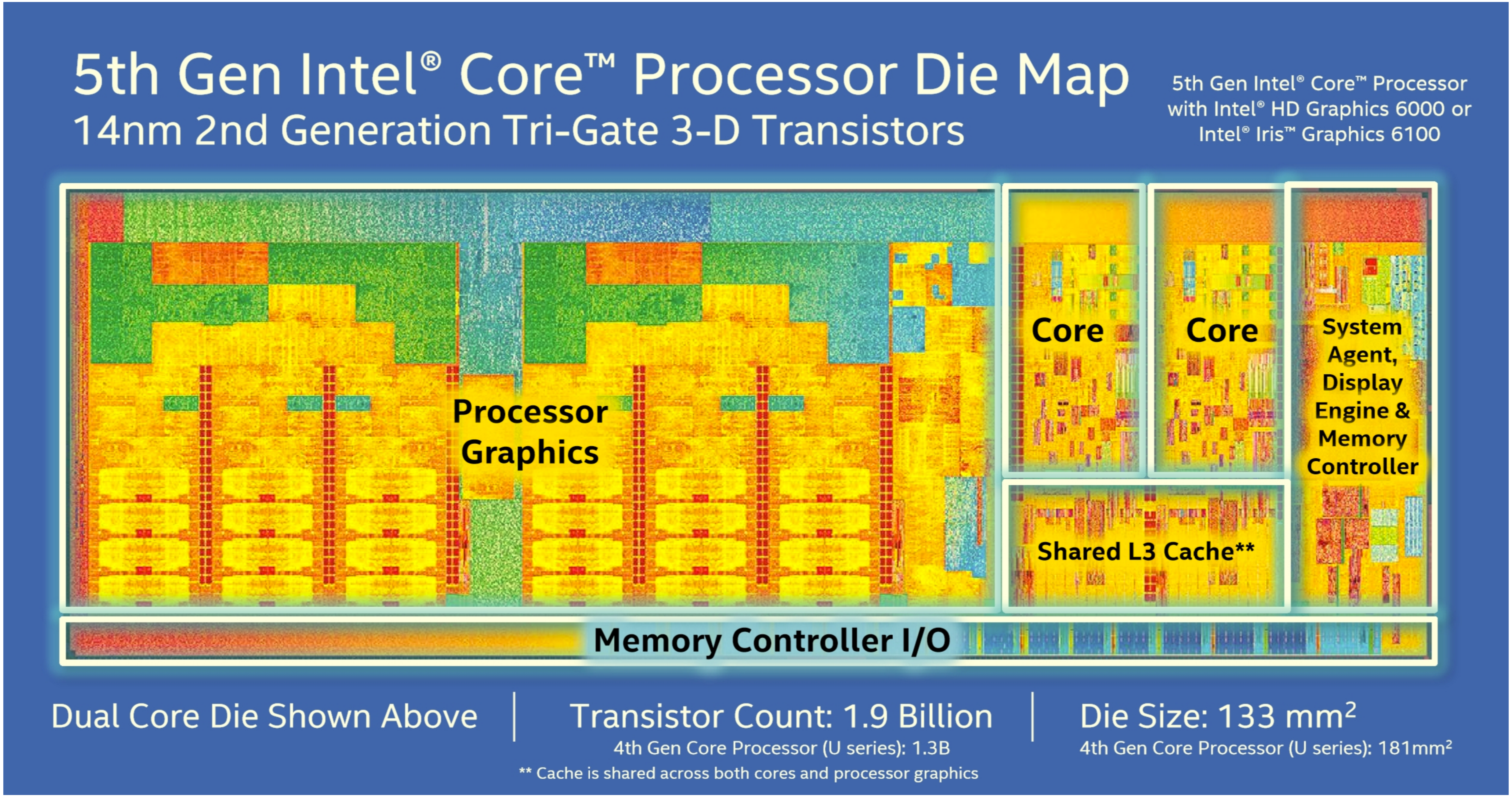


Convolutional Networks



Graph Convolutional Network

Modern hardware is heterogeneous and programming it is hard



5th Gen Intel® Core™ Processor Die Map

14nm 2nd Generation Tri-Gate 3-D Transistors

5th Gen Intel® Core™ Processor
with Intel® HD Graphics 6000 or
Intel® Iris™ Graphics 6100

Dual Core Die Shown Above | Transistor Count: 1.9 Billion | Die Size: 133 mm²

4th Gen Core Processor (U series): 1.3B

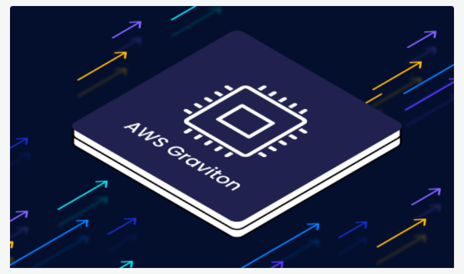
4th Gen Core Processor (U series): 181mm²

** Cache is shared across both cores and processor graphics

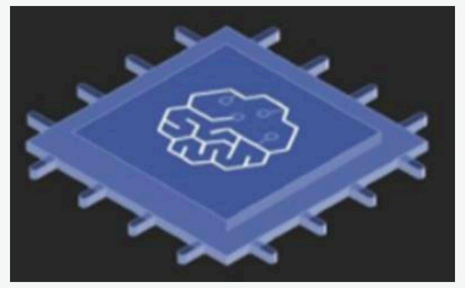
Hardware in the Clouds

aws

intel


Graviton

NVIDIA.

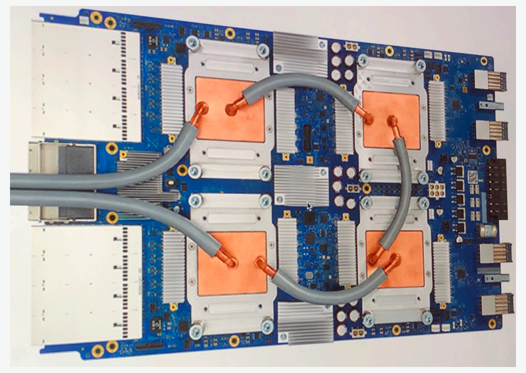

Inferentia

Trainium

Google Cloud

intel

NVIDIA.


TPU

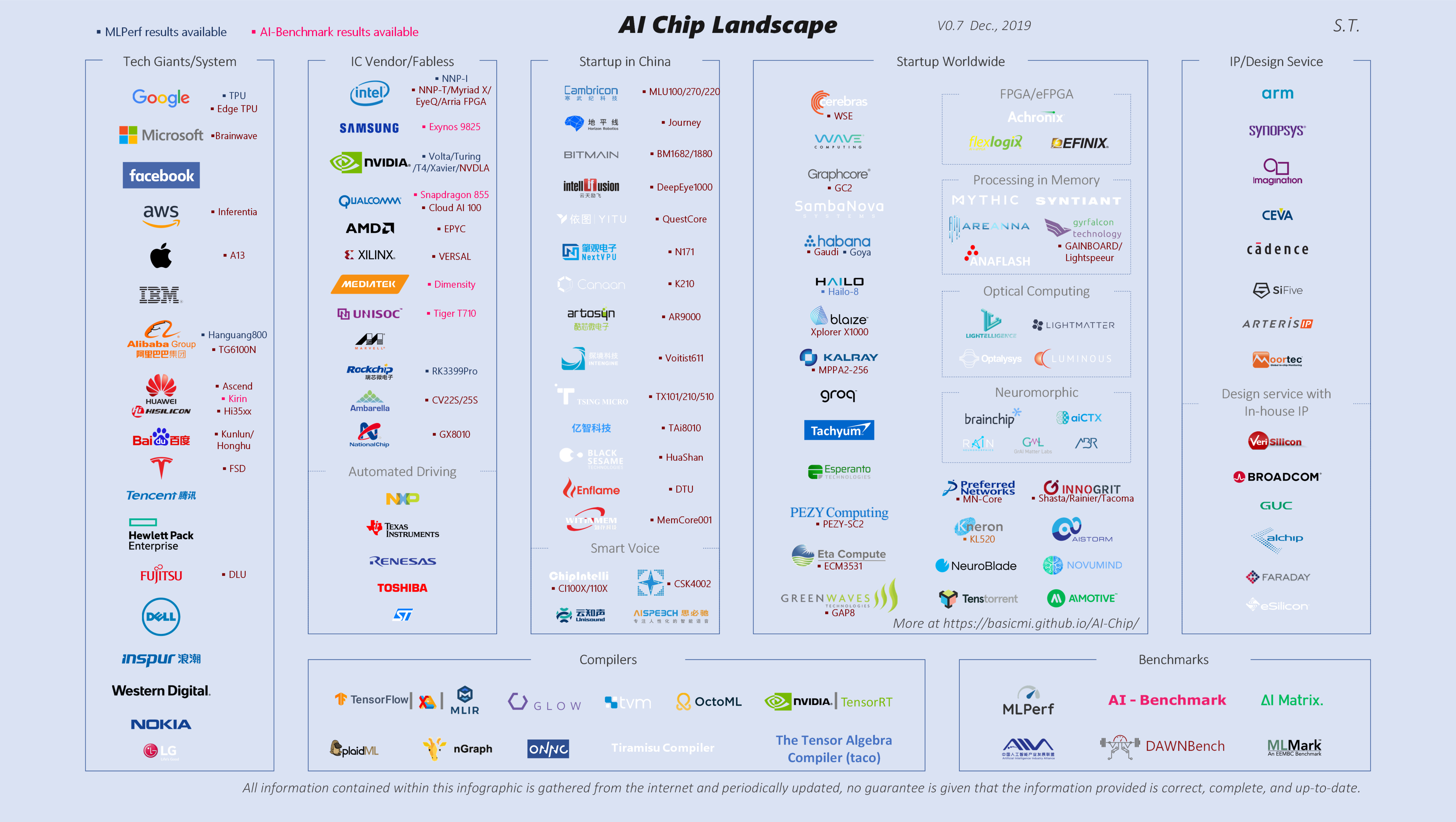
Microsoft Azure

intel

NVIDIA.

FPGAs

A lot of industry activity



AI Chip Landscape

V0.7 Dec., 2019 S.T.

▪ MLPerf results available ▪ AI-Benchmark results available

Tech Giants/System

- Google ▪ TPU ▪ Edge TPU
- Microsoft ▪Brainwave
- facebook
- aws ▪ Inferentia
- Apple ▪ A13
- IBM
- Alibaba Group 阿里巴巴集团 ▪ Hanguang800 ▪ TG6100N
- HUAWEI HISILICON ▪ Ascend ▪ Kirin ▪ Hi35xx
- Baidu 百度 ▪ Kunlun/ Honghu
- Tesla ▪ FSD
- Tencent 腾讯
- Hewlett Packard Enterprise
- FUJITSU ▪ DLU
- DELL
- inspur 浪潮
- Western Digital
- NOKIA
- LG

IC Vendor/Fabless

- intel ▪ NNP-I ▪ NNP-T/Myriad X/ EyeQ/Arria FPGA
- SAMSUNG ▪ Exynos 9825
- NVIDIA ▪ Volta/Turing /T4/Xavier/NVDLA
- QUALCOMM ▪ Snapdragon 855 ▪ Cloud AI 100
- AMD ▪ EPYC
- XILINX ▪ VERSAL
- MEDIATEK ▪ Dimensity
- UNISOC ▪ Tiger T710
- MARVELL
- Rockchip 瑞芯微电子 ▪ RK3399Pro
- Ambarella ▪ CV22S/25S
- NationalChip ▪ GX8010

Automated Driving

- NXP
- Texas Instruments
- RENESAS
- TOSHIBA
- ST

Startup in China

- Cambricon 寒武纪科技 ▪ MLU100/270/220
- 地平线 Horizon Robotics ▪ Journey
- BITMAIN ▪ BM1682/1880
- intellifusion 云天励飞 ▪ DeepEye1000
- 依图 YITU ▪ QuestCore
- 肇观电子 NextVPU ▪ N171
- Canaan ▪ K210
- artosyn 酷芯微电子 ▪ AR9000
- 探境科技 INTENGINE ▪ Voitist611
- TSING MICRO ▪ TX101/210/510
- 亿智科技 ▪ TAi8010
- BLACK SESAME TECHNOLOGIES ▪ HuaShan
- Enflame ▪ DTU
- WITINMEM 知存科技 ▪ MemCore001

Smart Voice

- ChipIntelli ▪ CI100X/110X
- ▪ CSK4002
- 云知声 Unisound
- AISPEECH 思必驰 专注人性化的智能语音

Startup Worldwide

- cerebras ▪ WSE
- WAVE COMPUTING
- Graphcore ▪ GC2
- SambaNova SYSTEMS
- habana ▪ Gaudi ▪ Goya
- HAILO ▪ Hailo-8
- blaize Xplorer X1000
- KALRAY ▪ MPPA2-256
- groq
- Tachyum
- Esperanto TECHNOLOGIES
- PEZY Computing ▪ PEZY-SC2
- Eta Compute ▪ ECM3531
- GREENWAVES TECHNOLOGIES ▪ GAP8

FPGA/eFPGA

- Achronix
- flexlogix
- EFINIX

Processing in Memory

- MYTHIC
- SYNTIANT
- AREANNA
- gyrfalcon technology ▪ GAINBOARD/ Lightspeeur
- ANAFLASH

Optical Computing

- LIGHTELLIGENCE
- LIGHTMATTER
- Optalysys
- LUMINOUS

Neuromorphic

- brainchip
- aiCTX
- RAIN NEUROMORPHICS
- GrAI Matter Labs
- ABR

- Preferred Networks ▪ MN-Core
- INNOGRIT ▪ Shasta/Rainier/Tacoma
- Kneron ▪ KL520
- AISTORM
- NeuroBlade
- NOVUMIND
- Tenstorrent
- AIMOTIVE

More at https://basicmi.github.io/AI-Chip/

IP/Design Sevice

- arm
- SYNOPSYS
- imagination
- CEVA
- cadence
- SiFive
- ARTERIS IP
- Moortec

Design service with In-house IP

- VeriSilicon
- BROADCOM
- GUC
- alchip
- FARADAY
- eSilicon

Compilers

- TensorFlow | XLA | MLIR
- GLOW
- tvm
- OctoML
- NVIDIA TensorRT
- plaidML
- nGraph
- ONNC
- Tiramisu Compiler
- The Tensor Algebra Compiler (taco)

Benchmarks

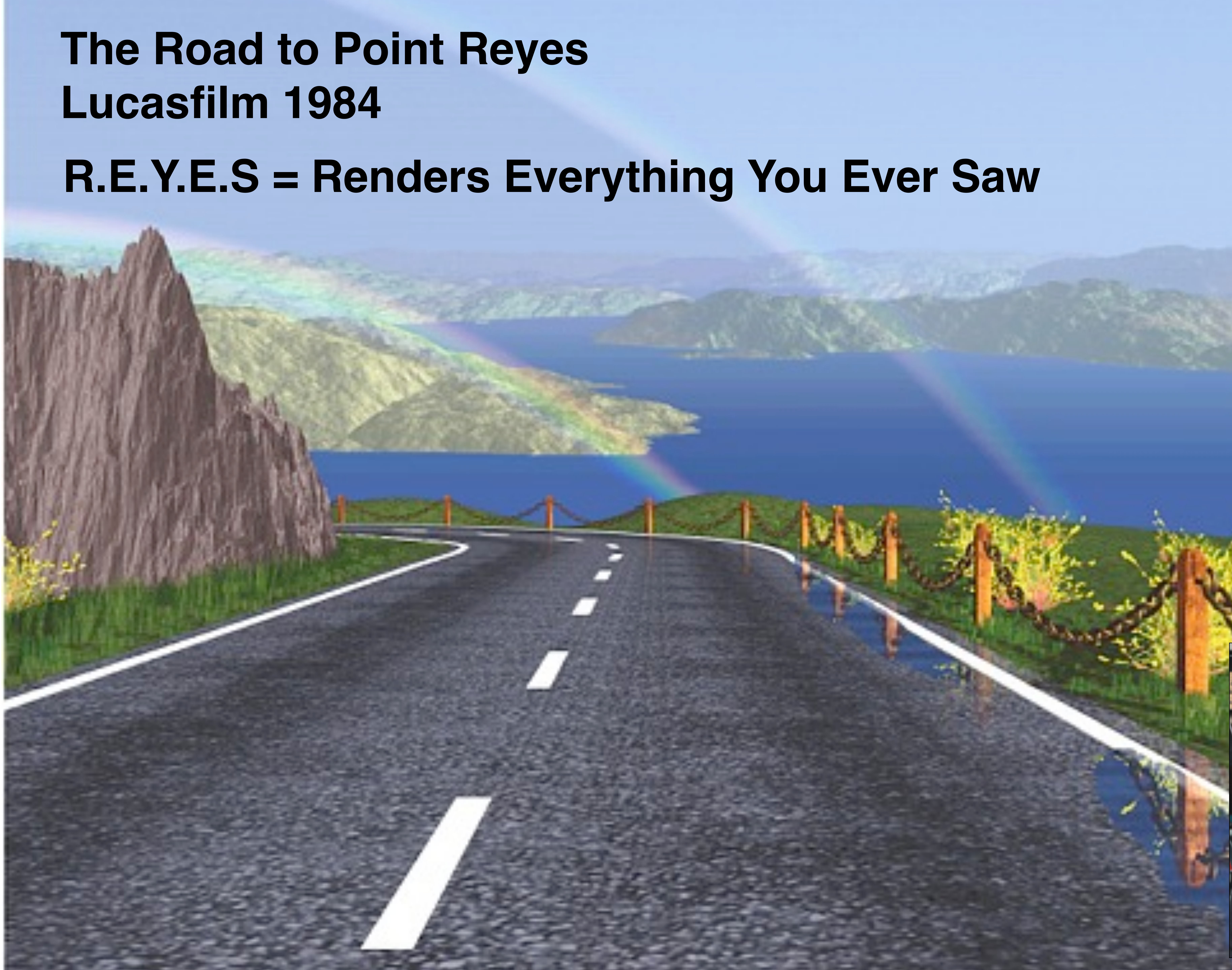
- MLPerf
- AI - Benchmark
- AI Matrix.
- AIIA 中国人工智能产业发展联盟 Artificial Intelligence Industry Alliance
- DAWNBench
- MLMark An EEMBC Benchmark

All information contained within this infographic is gathered from the internet and periodically updated, no guarantee is given that the information provided is correct, complete, and up-to-date.

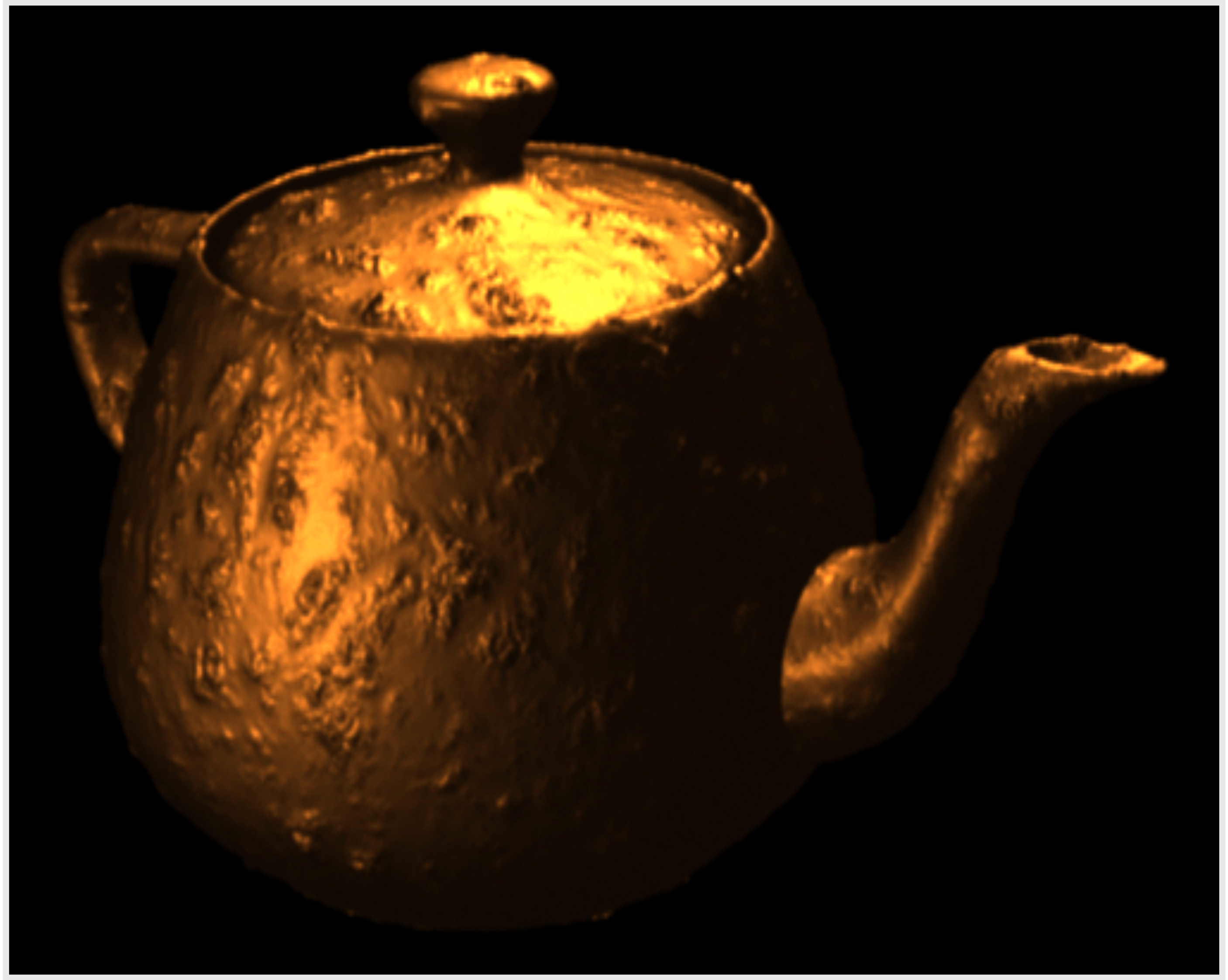
https://basicmi.github.io/AI-Chip/

The Road to Point Reyes
Lucasfilm 1984

R.E.Y.E.S = Renders Everything You Ever Saw



```
surface corrode(float Ks=0.4, Ka=0.1, rough=0.25) {
  float i, freq=1, turb=0;
  // compute fractal texture
  for( i=0; i<6; i++ ) {
    turb+=1/freq*noise(freq*P);
    freq*=2;
  }
  // perturb surface
  P -= turb * normalize(N);
  N = faceforward(normalize(calculatenormal(P)));
  // compute reflection and final color
  Ci = Cs*(Ka*ambient()+Ks*specular(N,I,rough));
}
```



Little Languages (DSLs)

Jon Bentley, CACM 29(8), 1986

Defining “little” is harder; it might imply that the first-time user can use this system in an hour or master the language in a day, or perhaps the first implementation took just a few days. In any case, a little language is specialized to a particular problem domain and does not include many features found in conventional languages.

UNIX "DSLs"

bash, csh - shell programming

awk - processing strings

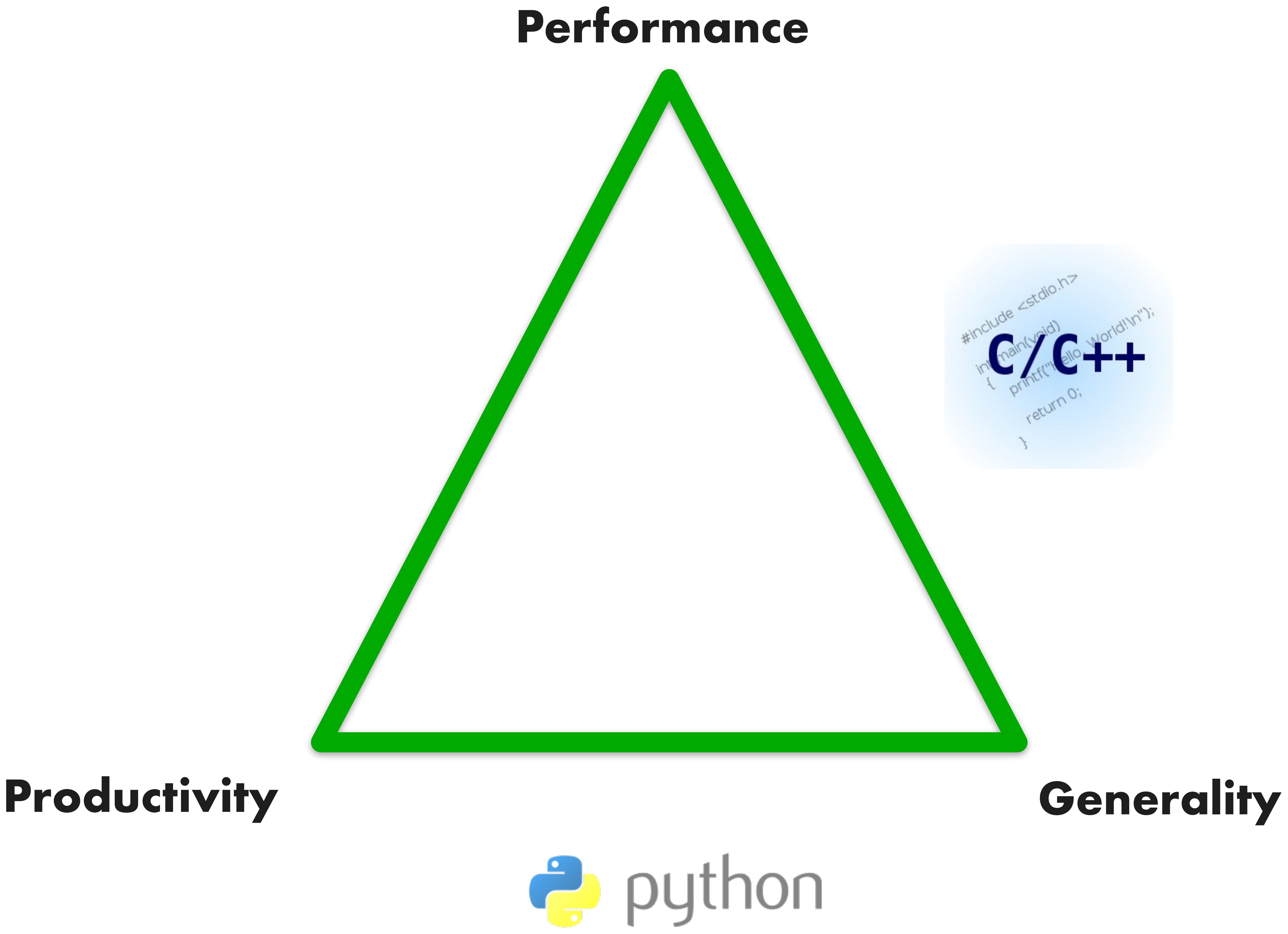
sed - regular expressions

troff, pic, tbl, eqn, …

printf formatting

…

Programming Languages

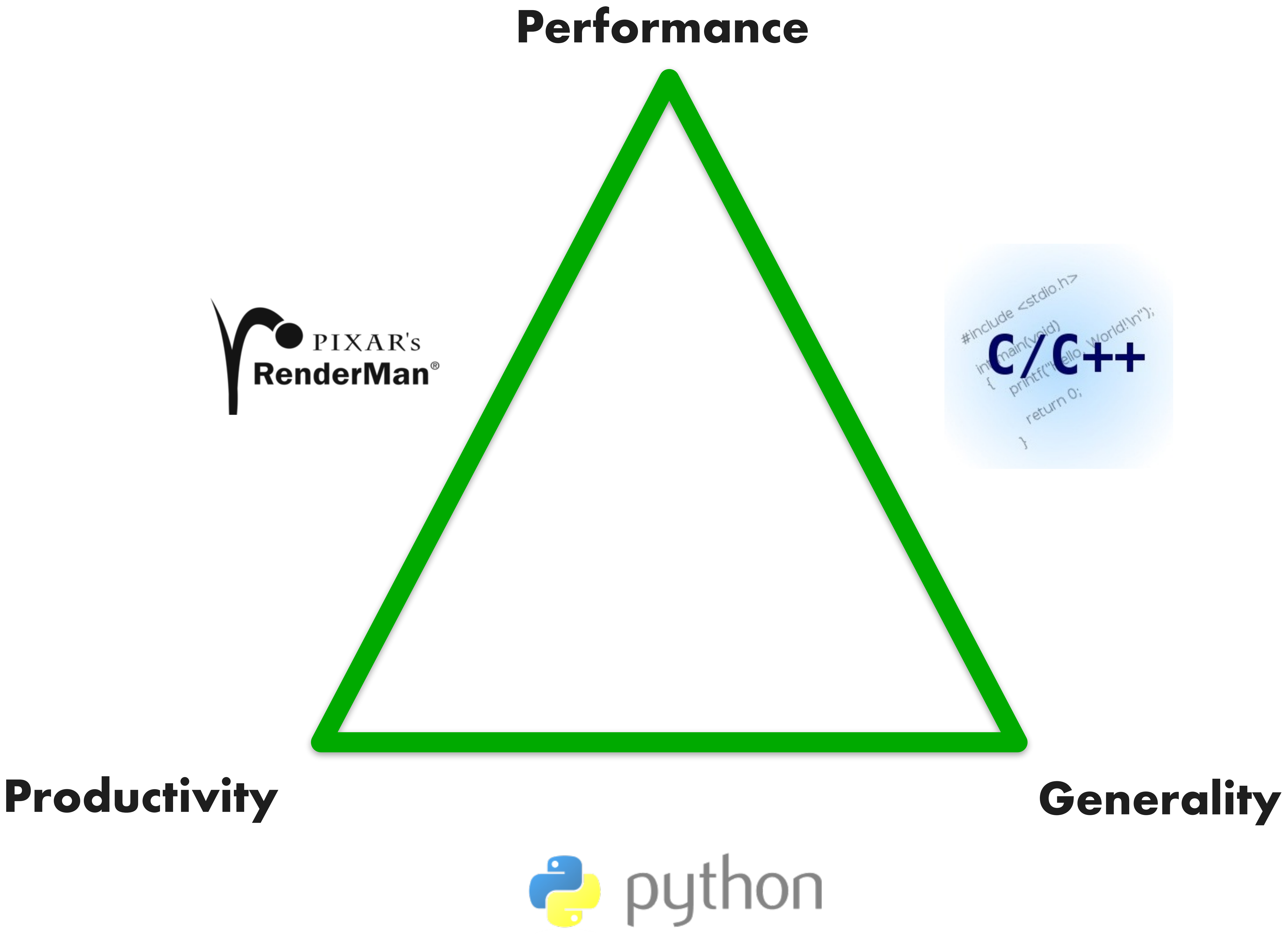


Performance

Productivity

Generality

Domain-Specific Languages



Performance

Productivity

Generality

Graphics Libraries

```
glPerspective(45.0);
for( … ) {
    glTranslate(1.0,2.0,3.0);
    glBegin(GL_TRIANGLES);
        glVertex(…);
        glVertex(…);
        …
    glEnd();
}
glSwapBuffers();
```

OpenGL “Grammar”

```
<Scene> = <BeginFrame> <Camera> <World>
<EndFrame>

<Camera> = glMatrixMode(GL_PROJECTION)
<View>
<View> = glPerspective | glOrtho

<World> = <Objects>*
<Object> = <Transforms>* <Geometry>
<Transforms> = glTranslatef | glRotatef | …
<Geometry> = glBegin <Vertices> glEnd
<Vertices> = [glColor] [glNormal] glVertex
```

Advantages

Productivity

- Graphics library is easy to use

Portability

- Runs on wide range of GPUs

Advantages

Productivity

Portability

Performance

- Vertices/Fragments are independent
- Rasterization can be done in hardware
- Textures are read-only; texture filtering hw
- Specialized scheduler for pipeline
- …
- Allows for super-optimized implementations

Advantages

Productivity

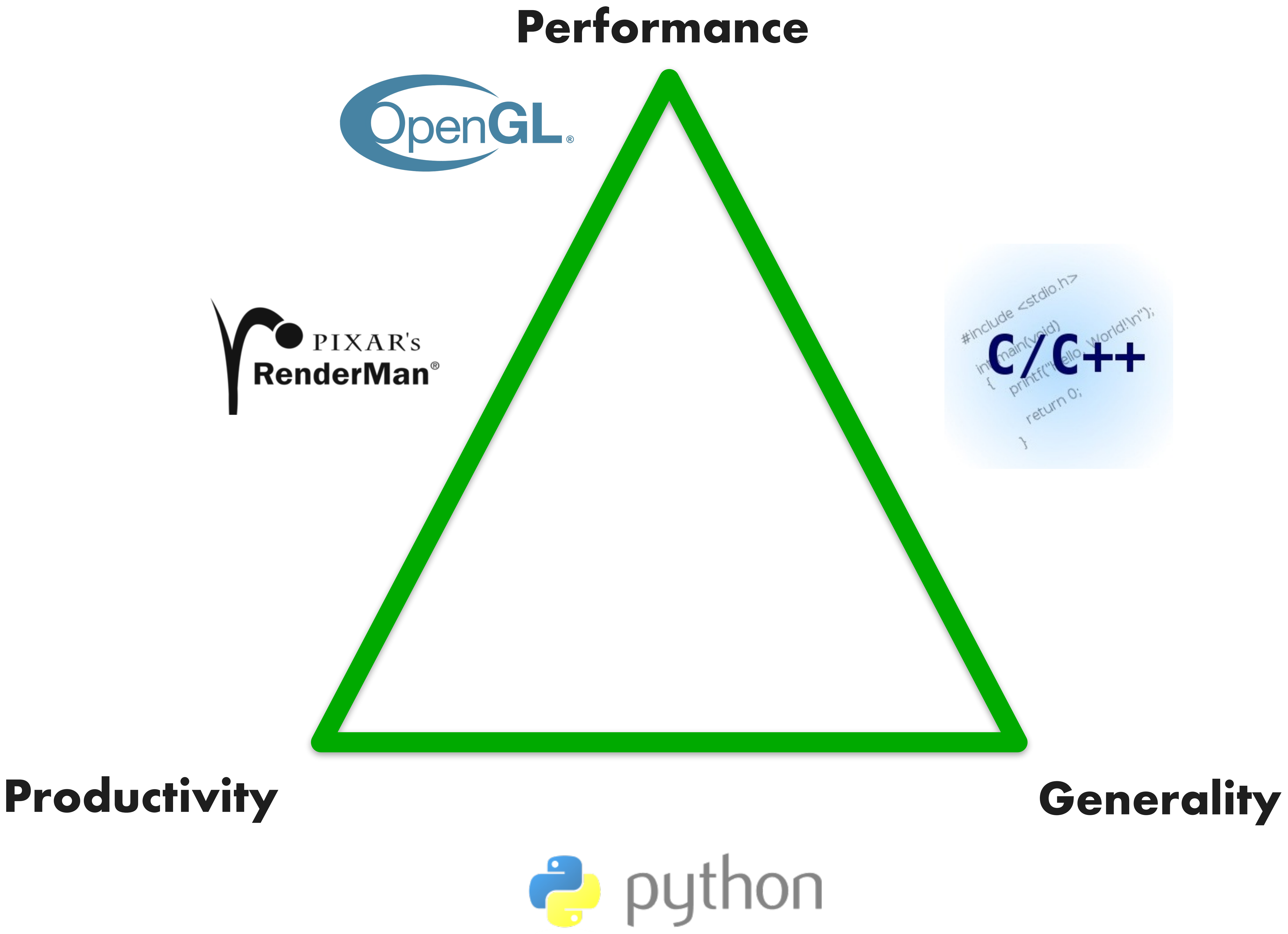
Portability

Performance

Encourage innovation

- Allows vendors to radically optimize hardware architecture to achieve efficiency
- Allows vendors to introduce new low-level programming models and abstractions

Domain-Specific Languages



Performance

Productivity

Generality

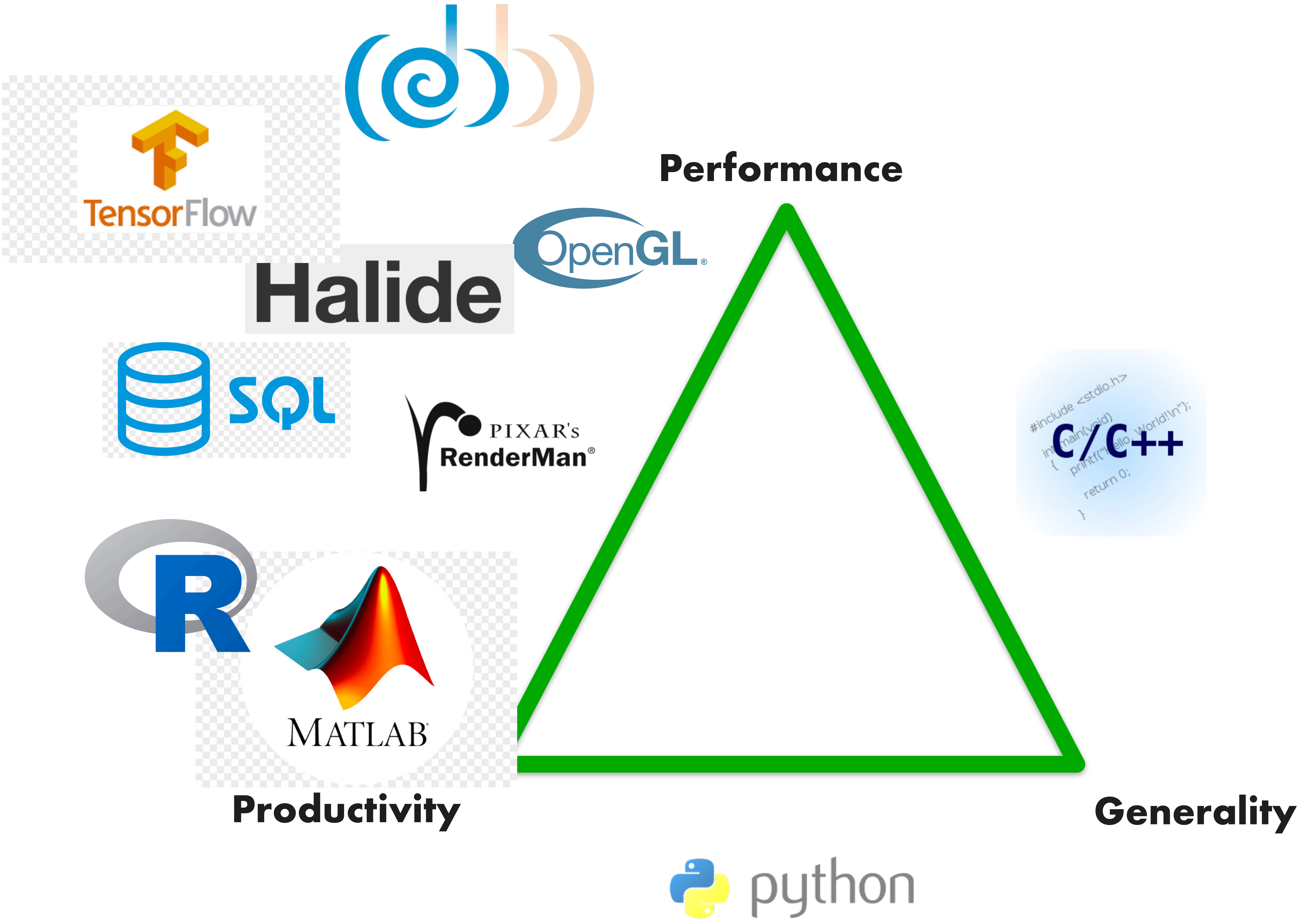
Definition: Domain-Specific

Definition: A language or library that exploits domain knowledge for productivity and performance

Widely used in many application areas

- matlab / R
- SQL / map-reduce / Microsoft’s LINQ
- TensorFlow, pytorch

Domain-Specific Languages



Performance

Productivity

Generality

Why DSLs Work

Advantages

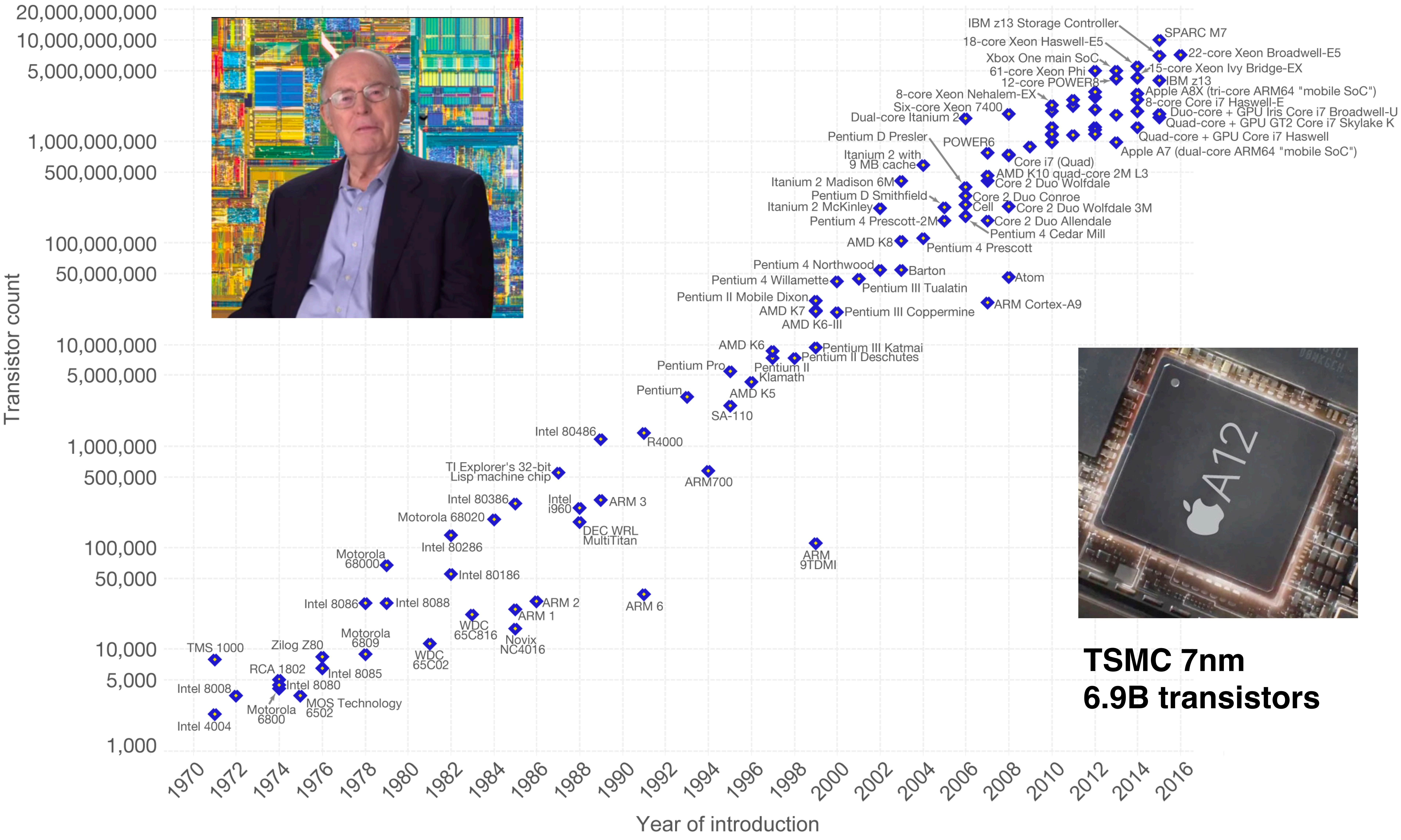
- Add the semantics of the domain
    - High-level program transformations
- Restrict programming language
    - Less-general computations
    - Guarantee static analysis
- Known parallelization strategies
    - Someone has shown how to robustly do it

=> Tractable

Moore's Law – The number of transistors on integrated circuit chips (1971-2016)

Our World in Data

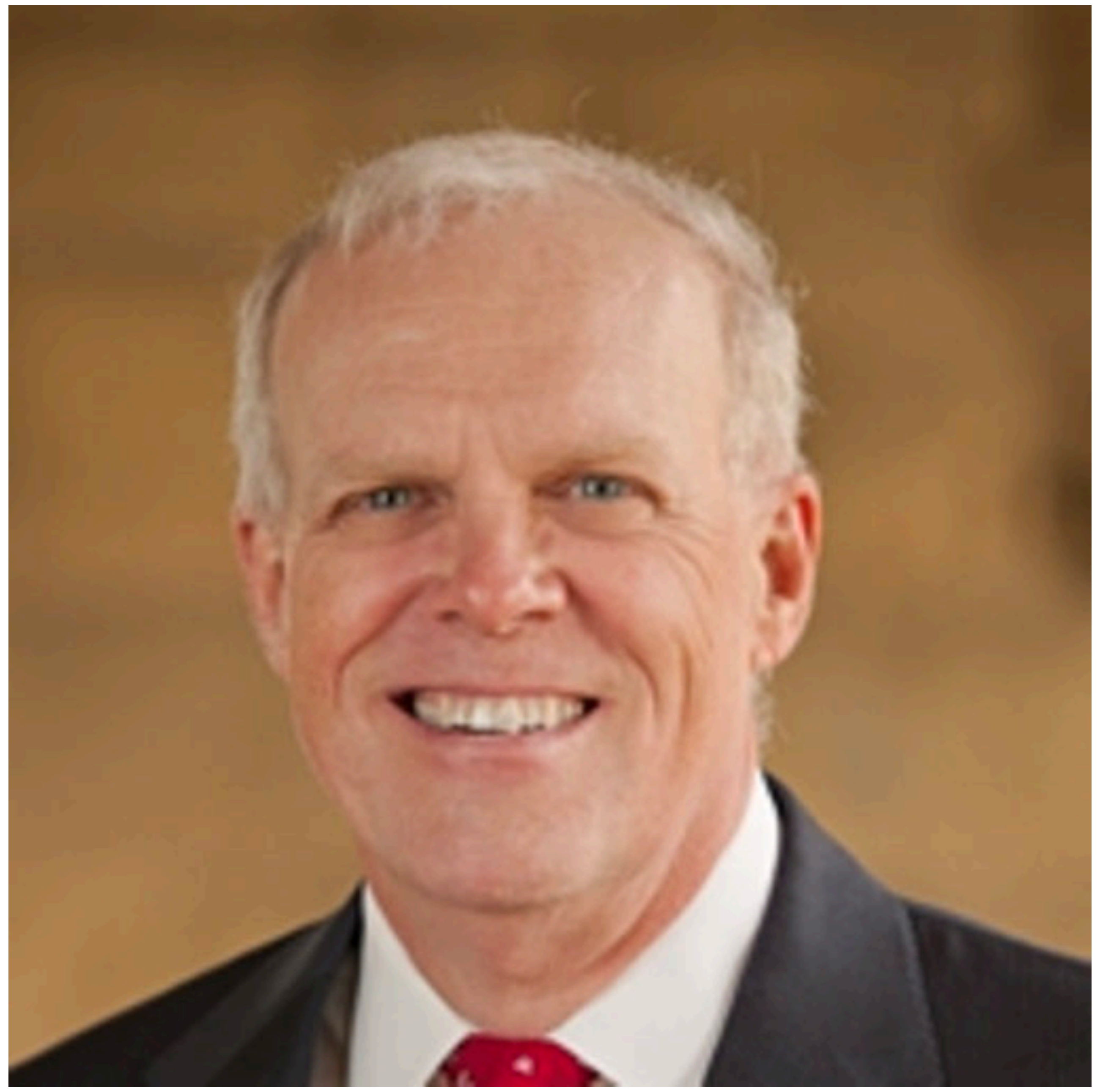
Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.



TSMC 7nm
6.9B transistors

Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

A New Golden Age for Computer Architecture: Domain-Specific Hardware/Software Co-Design

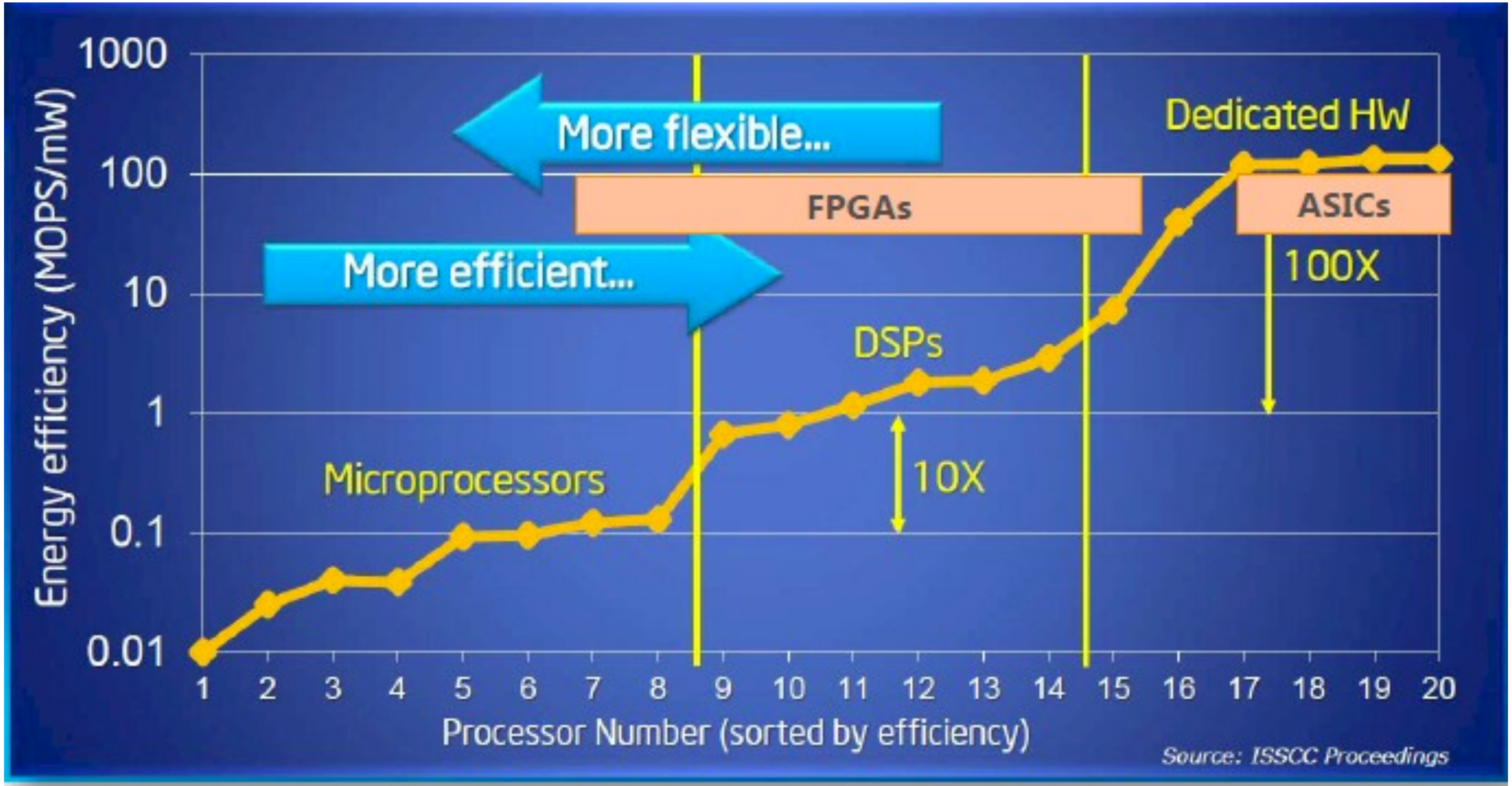


John Hennessy

David Patterson

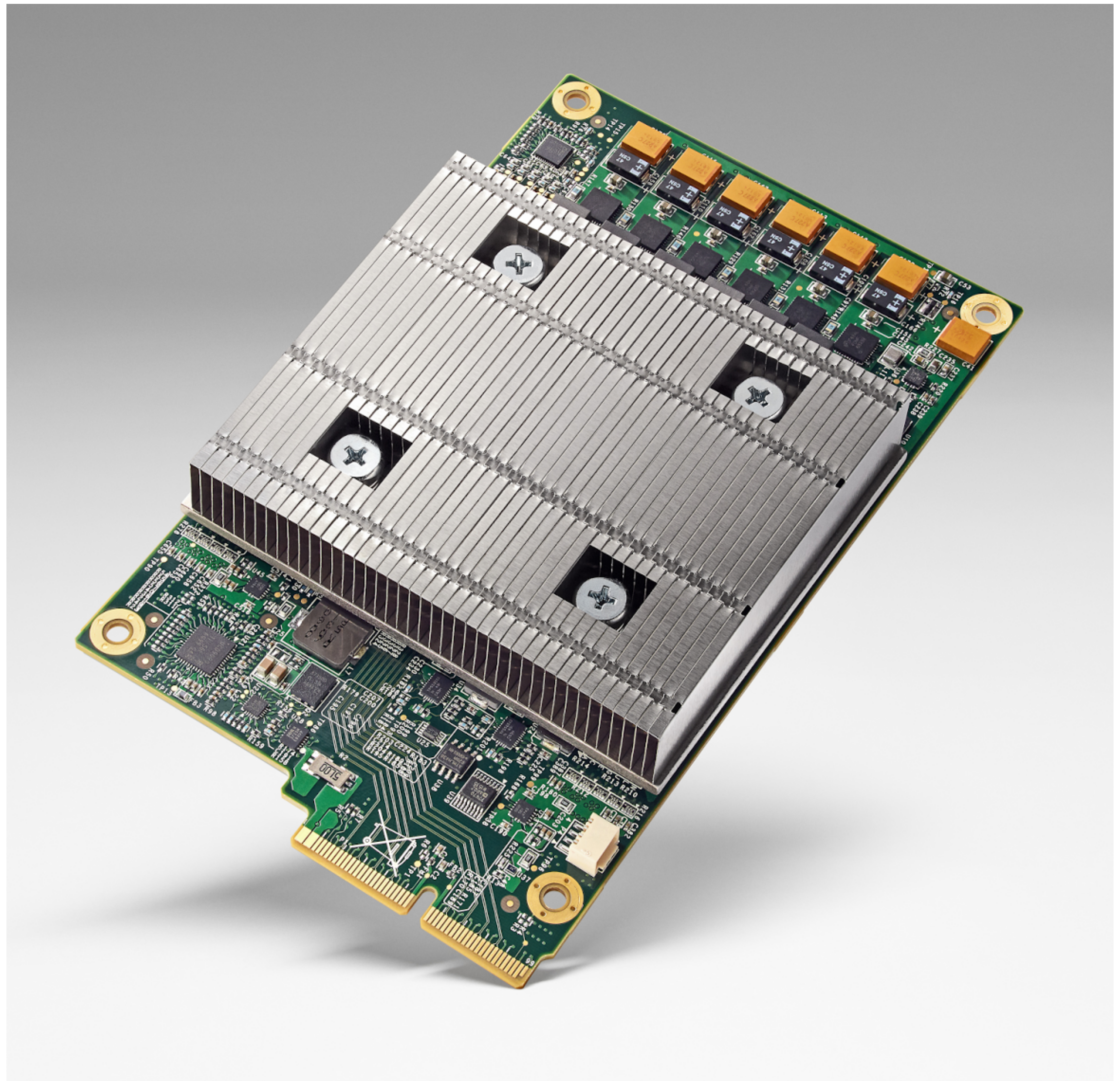
2017 Turing Award

Large efficiency gains with domain-specific architectures

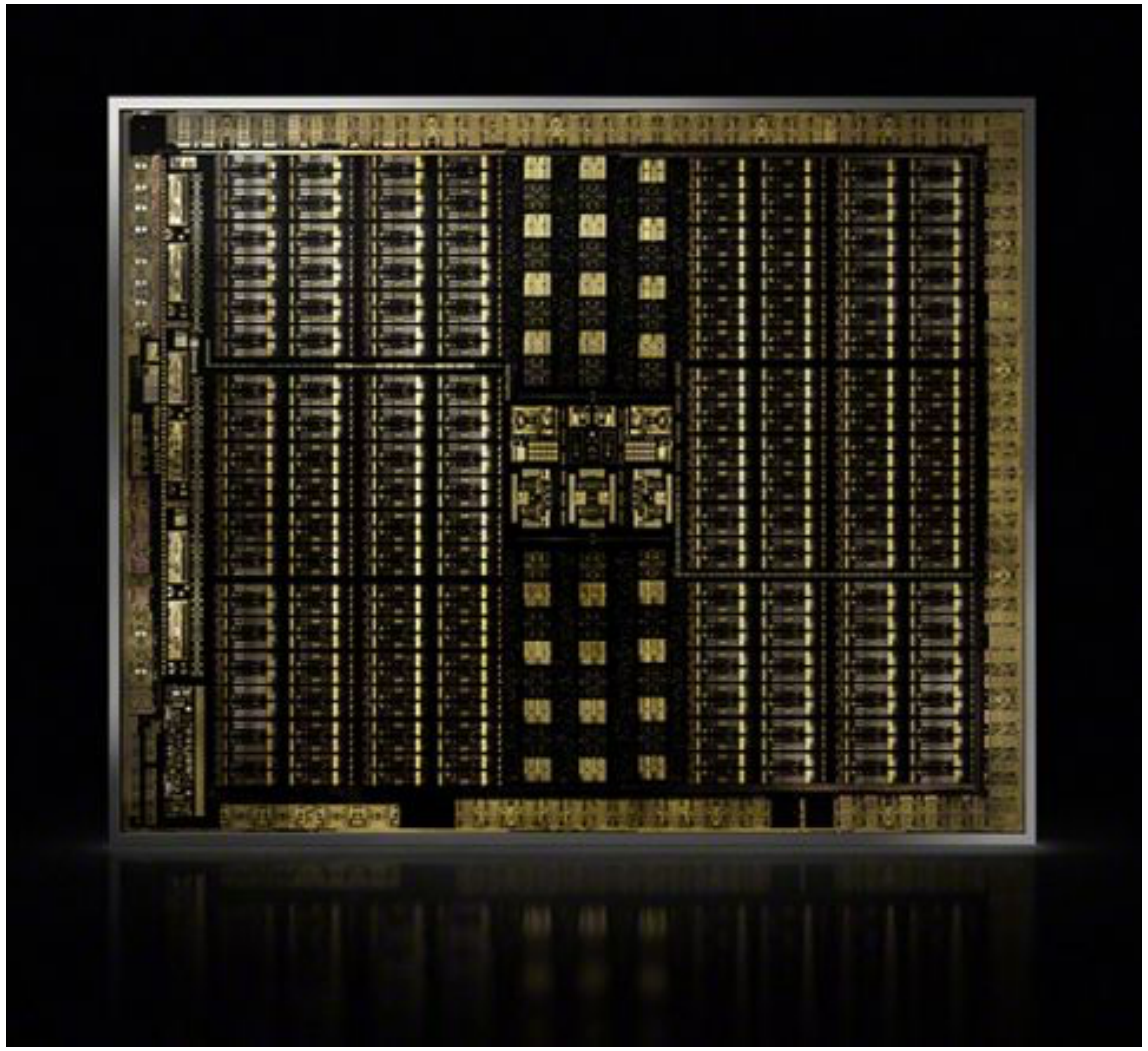


Source: Bob Broderson, Berkeley Wireless group

Domain-Specific Architectures



Google
Tensor Processing Unit



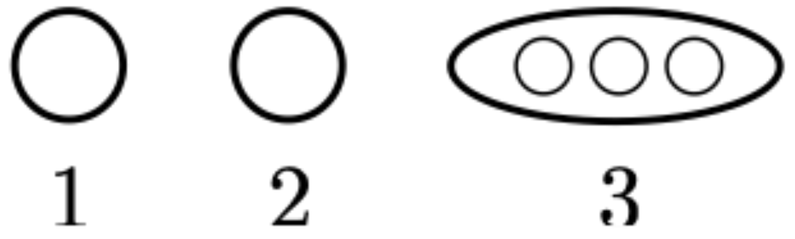
NVIDIA
Turing Architecture

Collection-Oriented Languages

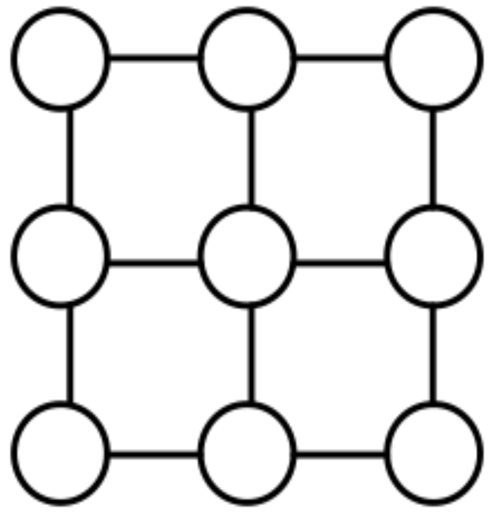
Lists
Lisp M58

Sets
SETL S70

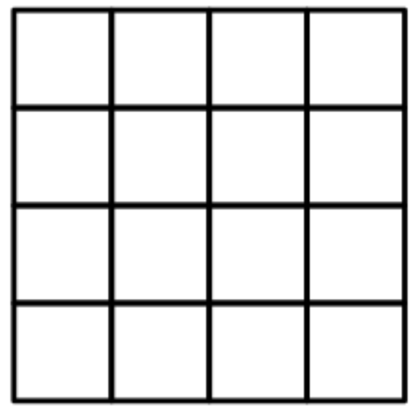
Nested Sequences
NESL B94



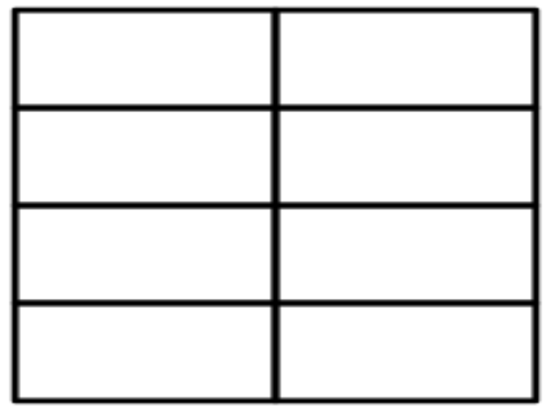
Grids
Sejits S09, Halide



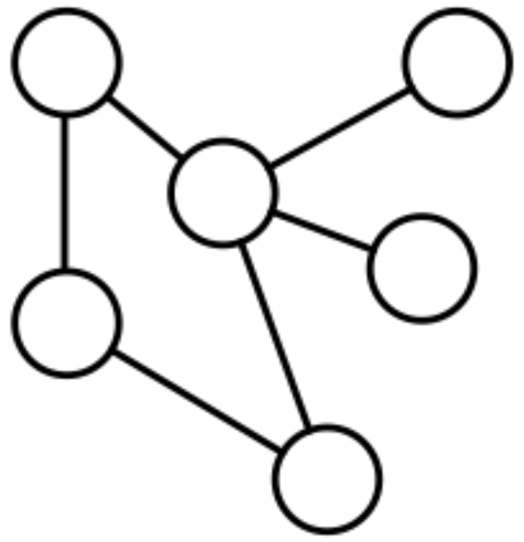
Arrays
APL I62



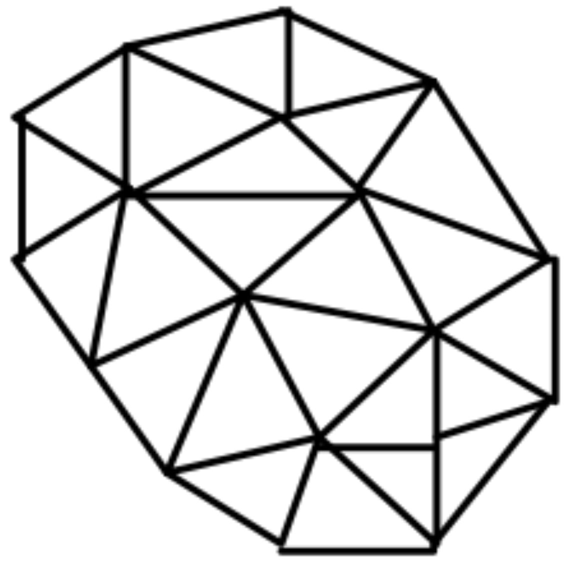
Relations
Relational Algebra C70,



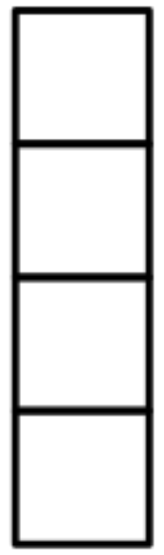
Graphs
GraphLab L10



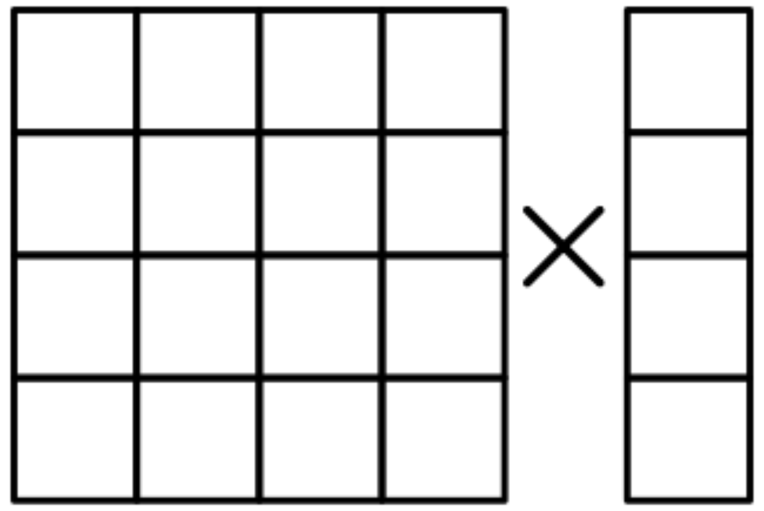
Meshes
Liszt D11



Vectors
Vector Model B90



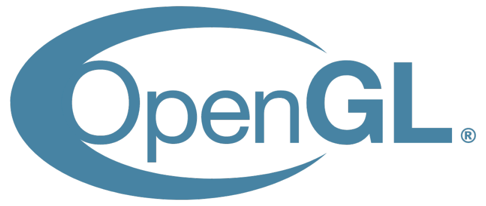
Matrices and Tensors
Matlab M79, taco K17



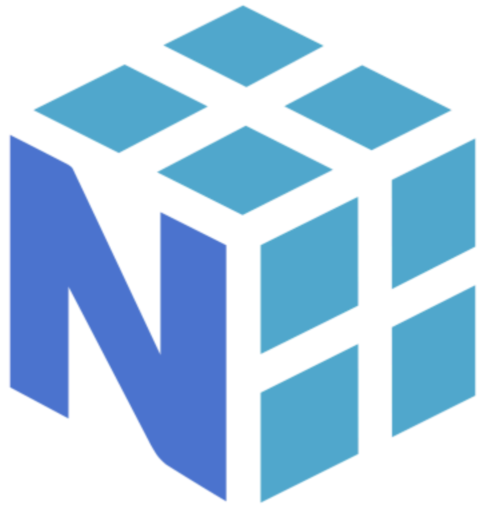
A collection-oriented programming model provides collective operations on some collection/abstract data structure

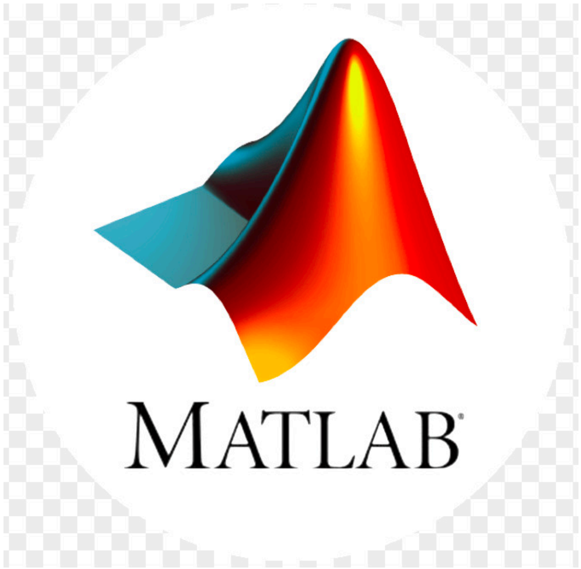
Modern Domain-Specific Languages/Compilers

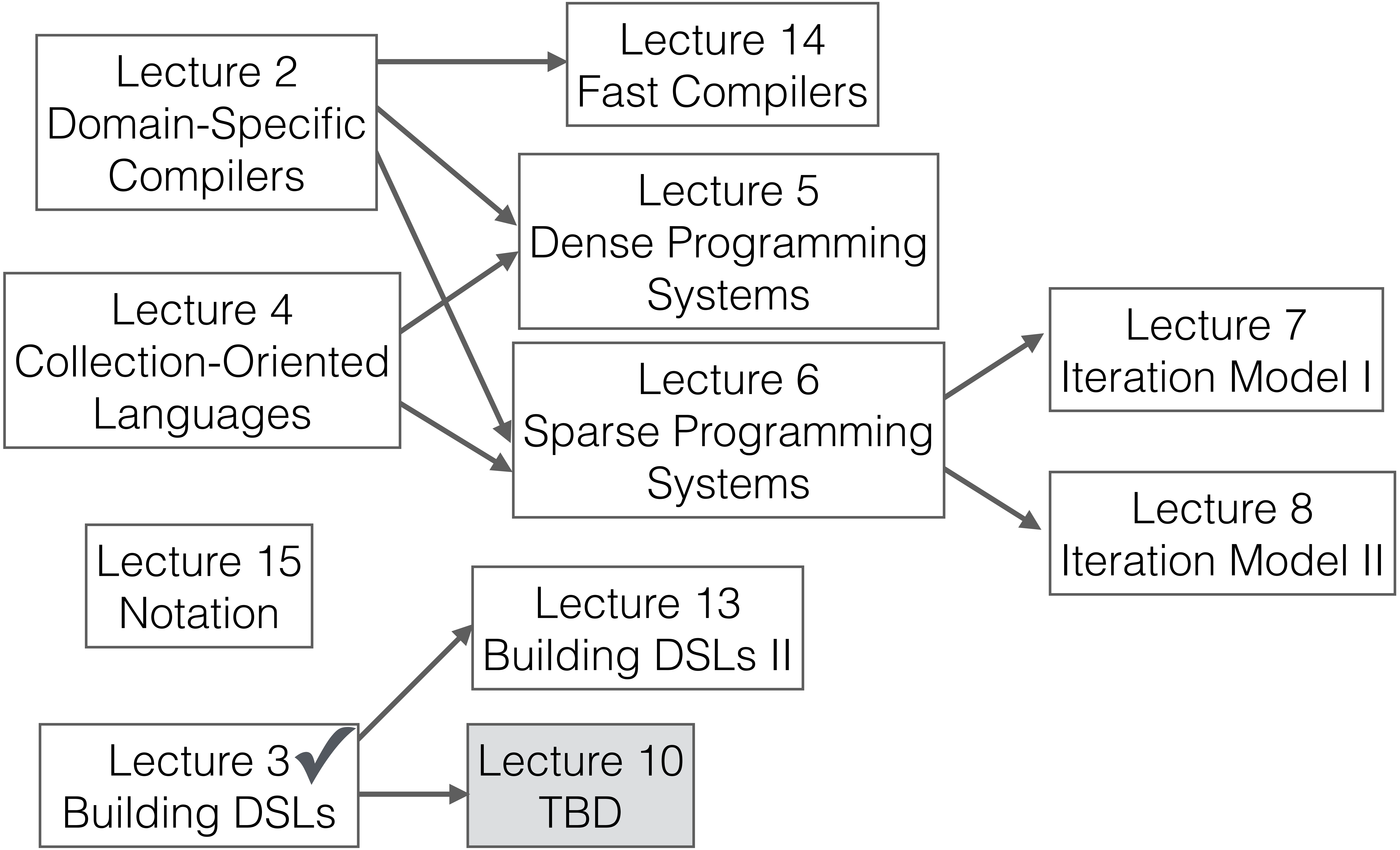
TensorFlow



PIXAR's
RenderMan






Lecture 2
Domain-Specific
Compilers
Lecture 14
Fast Compilers
Lecture 5
Dense Programming
Systems
Lecture 4
Collection-Oriented
Languages
Lecture 6
Sparse Programming
Systems
Lecture 7
Iteration Model I
Lecture 8
Iteration Model II
Lecture 15
Notation
Lecture 13
Building DSLs II
Lecture 3 ✓
Building DSLs
Lecture 10
TBD