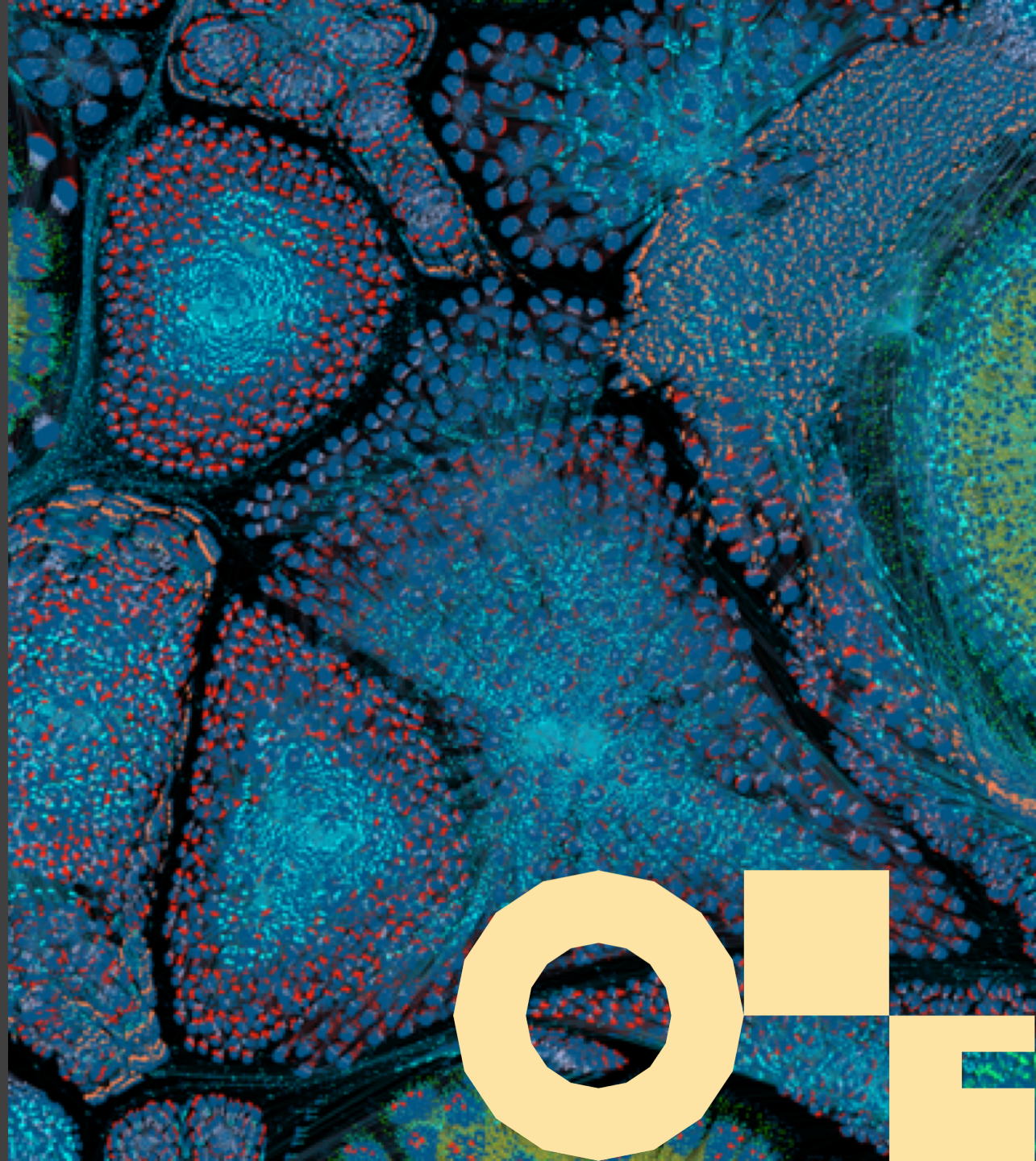
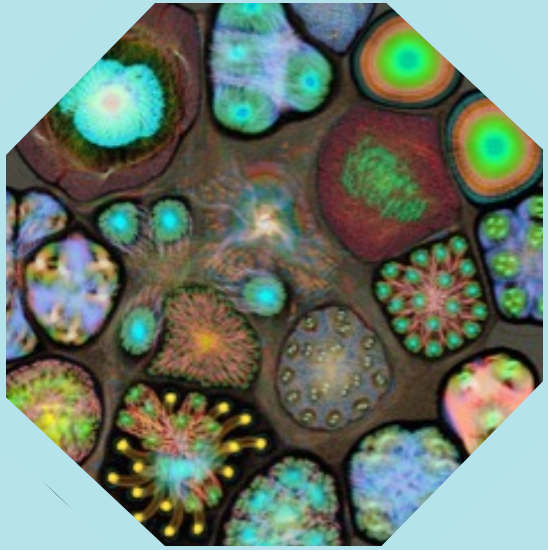
GRAPHCORE



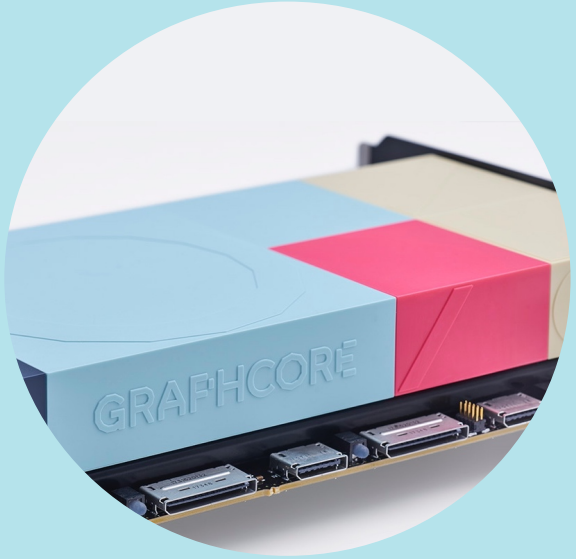
ABOUT US...

Technology



Processors and software solutions designed for AI

Products

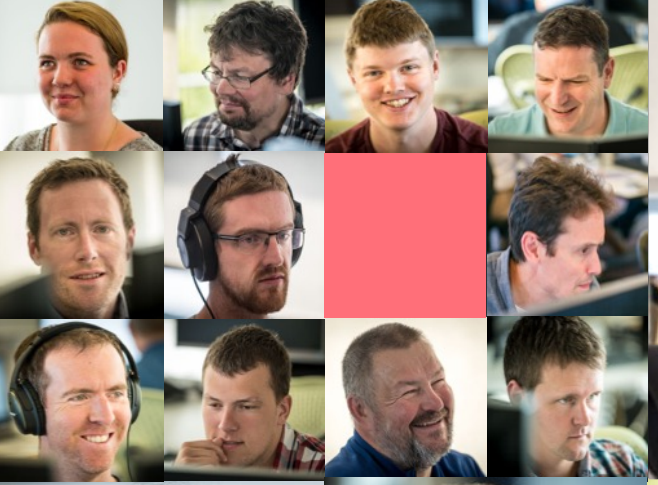


IPU-Processor PCIe Cards and Poplar® software stack

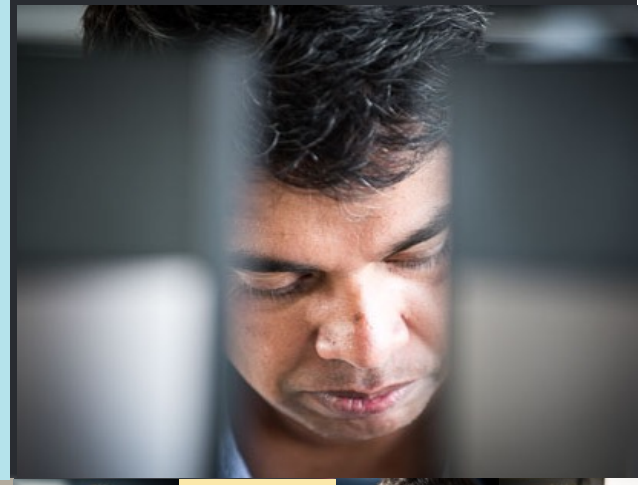
Investors

pitango VENTURE CAPITAL
draperesprit
Foundation CAPITAL
Merian GLOBAL INVESTORS
Amadeus Capital
SOFINA
SEQUOIA
Microsoft
ATOMICO
BMW i VENTURES
DELL
BOSCH
SAMSUNG

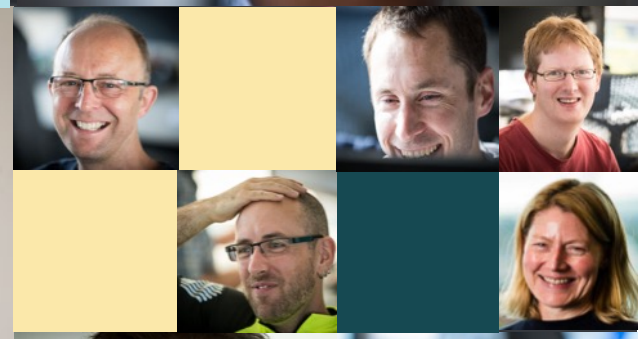
>$310m in funding



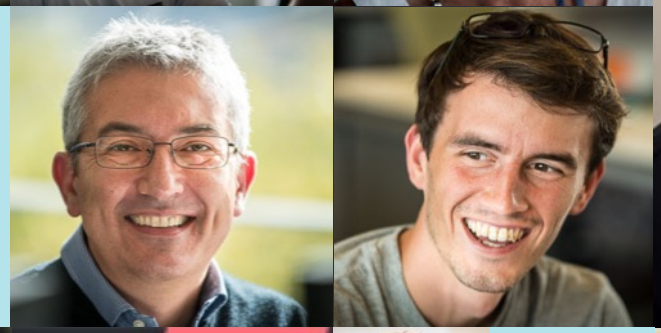

BRISTOL,
LONDON,
PALO ALTO
AUSTIN, SEATTLE
OSLO, BEIJING



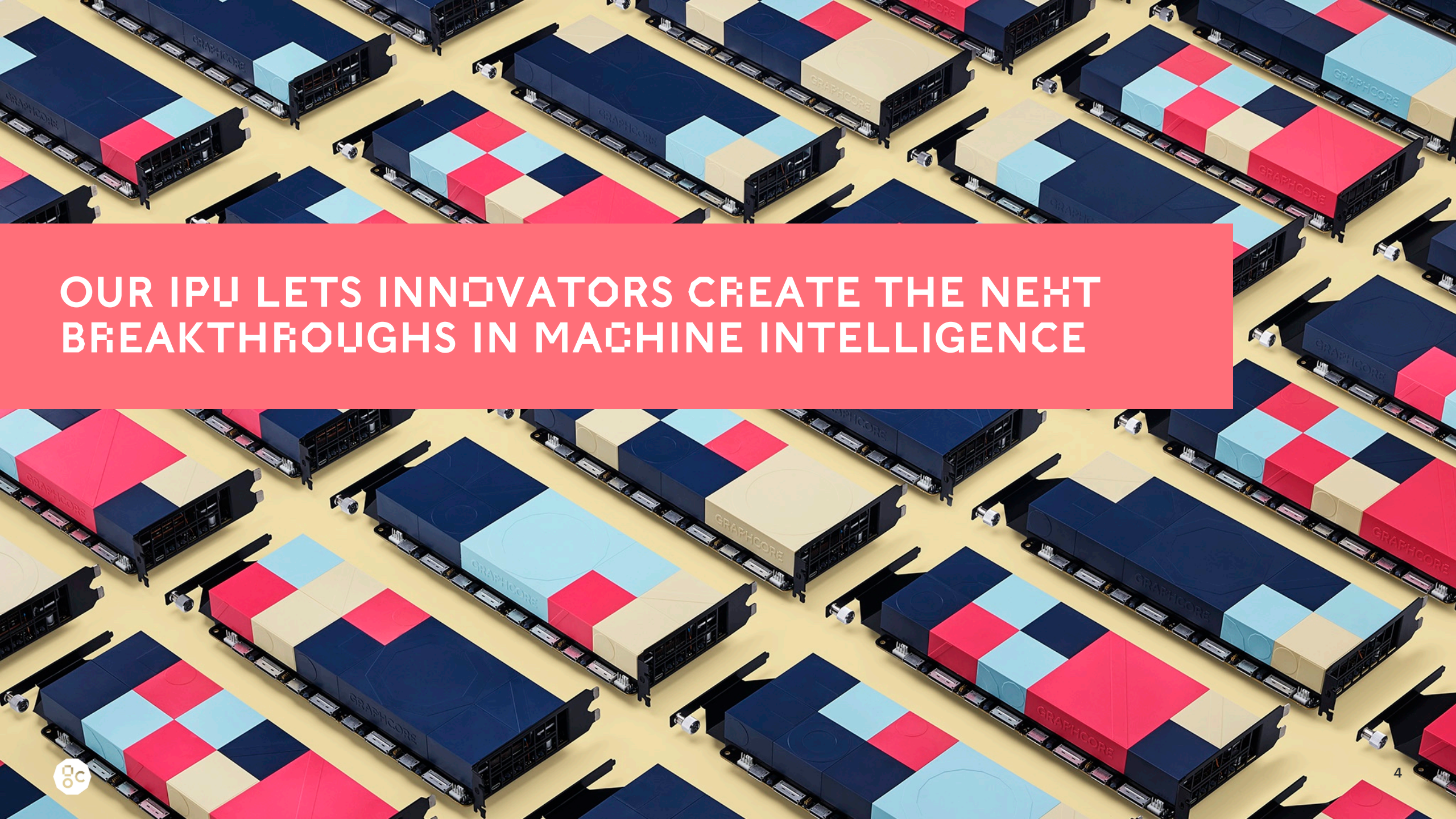




200+ TEAM &
GROWING
FAST







OUR IPU LETS INNOVATORS CREATE THE NEXT
BREAKTHROUGHS IN MACHINE INTELLIGENCE

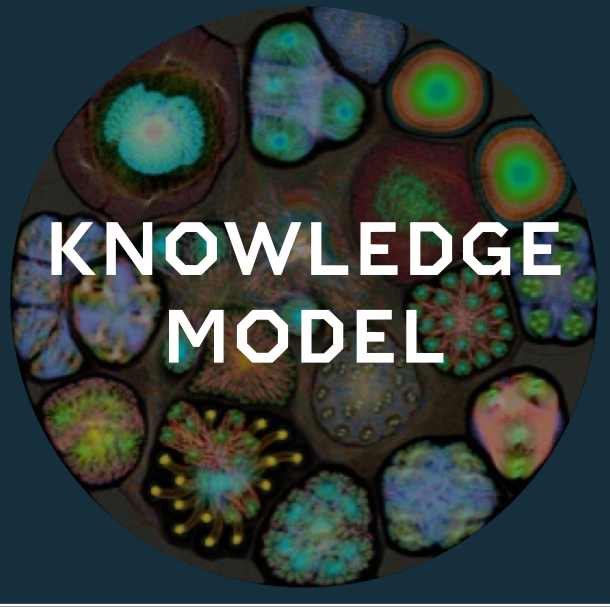
A COMPLETELY NEW WORKLOAD

Intelligence processing is highly parallel probabilistic optimization on a graph structure

TRAINING DATA / SENSOR DATA

Learn parameters (and model) from data

TRAINING & INFERENCE PROCESSOR

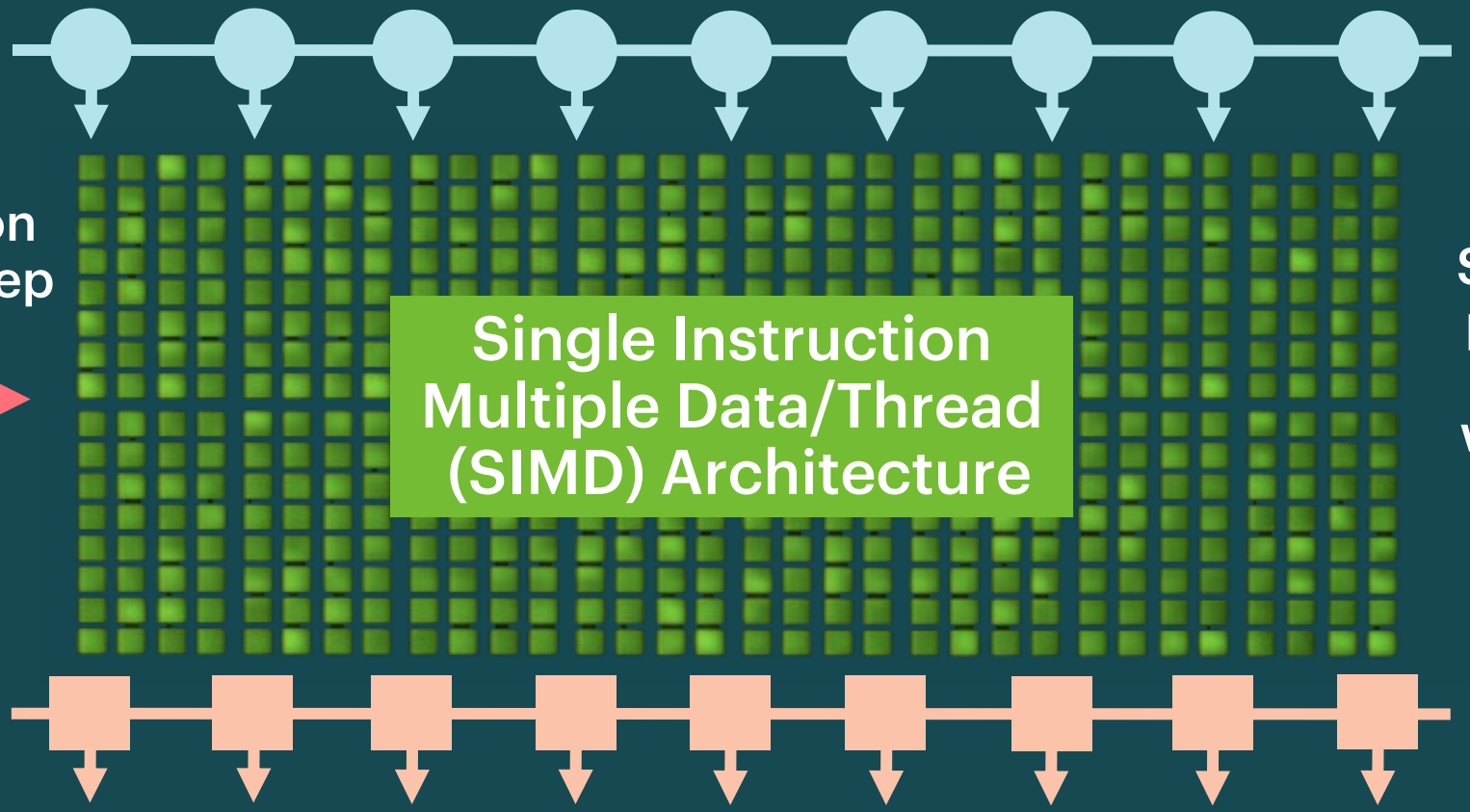


RESULT

Infer outputs using knowledge model

TODAY'S PARALLEL MACHINES ARE INFLEXIBLE

Instruction operates on long data vector
(typically 32x32 bits or 64x16 bits)



All units perform the same Instruction on each program step

Small number of processors *(~80)* each with a wide vector unit

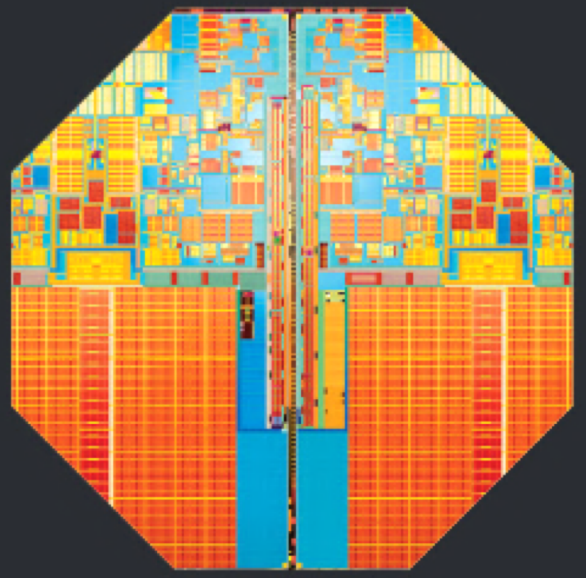
Only efficient for large blocks of dense data / large mini-batches

MEMORY BANDWIDTH IS LIMITING PERFORMANCE

If 10x faster...

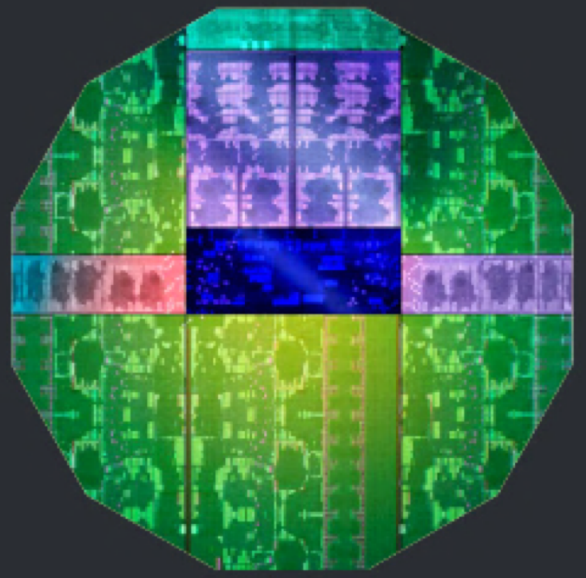
...10x faster how?

A NEW PROCESSOR IS REQUIRED



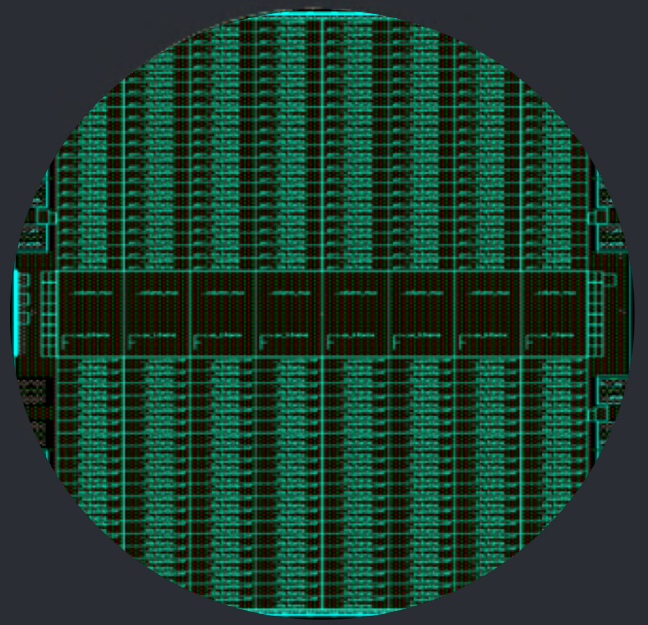
CPU

Apps and Web
Scalar



GPU

Graphics and HPC
Vector



IPU

Machine Intelligence
Graph

IPU ADVANTAGE

MASSIVE PERFORMANCE LEAP

- up to 100x faster on training and inference
- model held inside the processor
- 100x memory bandwidth

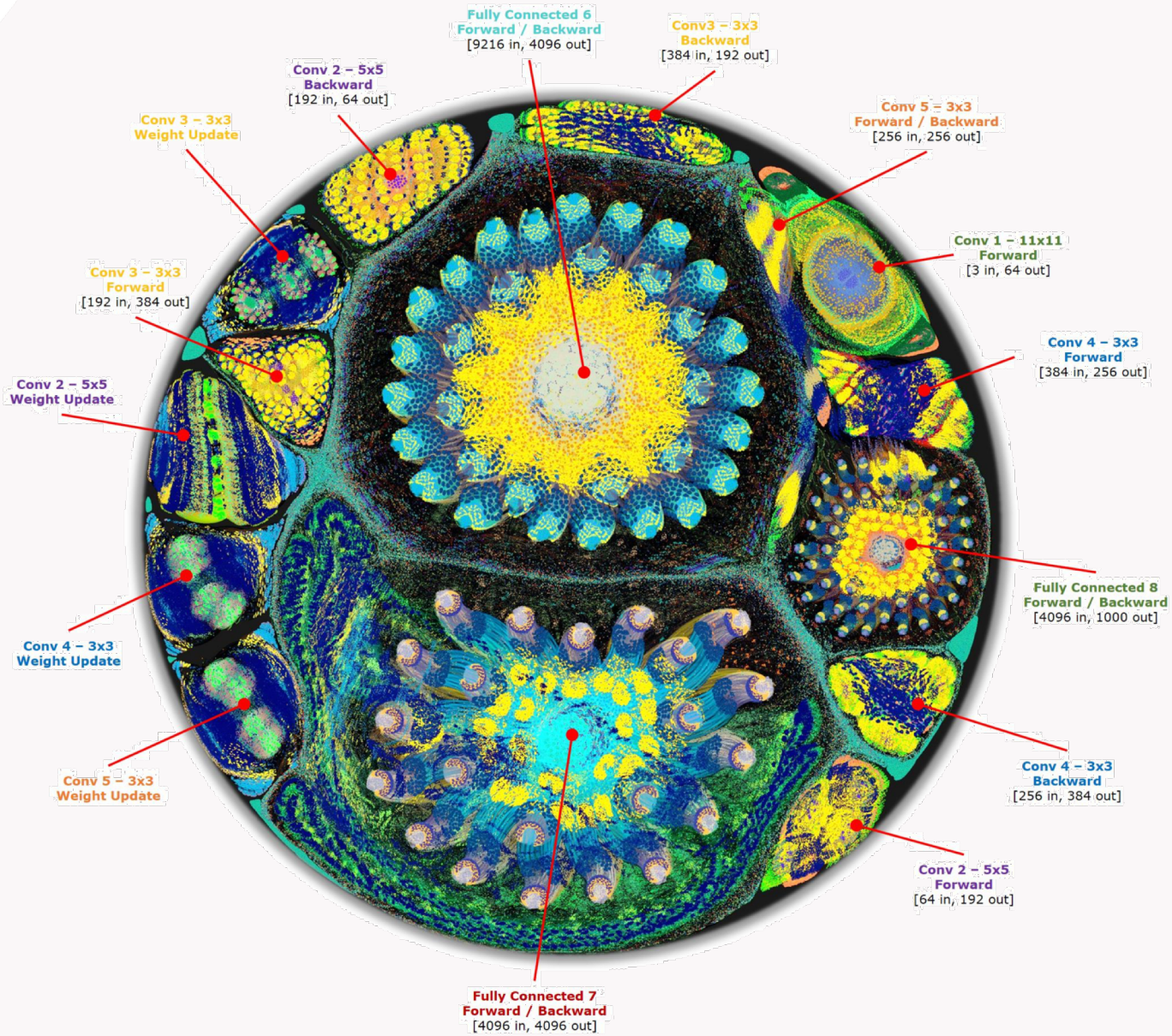
MUCH MORE FLEXIBLE

- every network type supported efficiently
- latency reduced by over 10x

EASIER TO USE

- seamless ML framework support
- Poplar® software stack

ALEXNET network visualization from POPLAR™



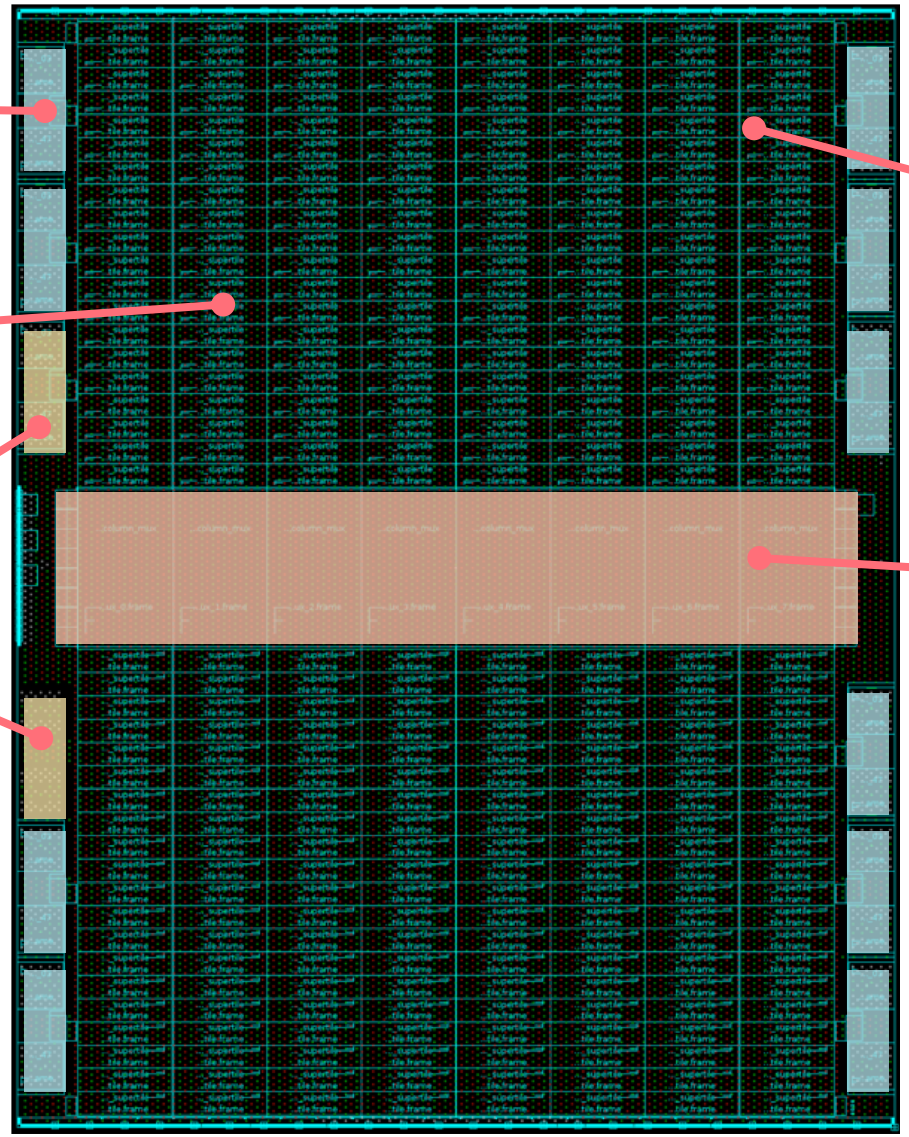
COLOSSUS GC2

The world's most complex processor chip with 23.6 billion transistors

10x IPU-LINKS™
320GB/s chip-to-chip bandwidth

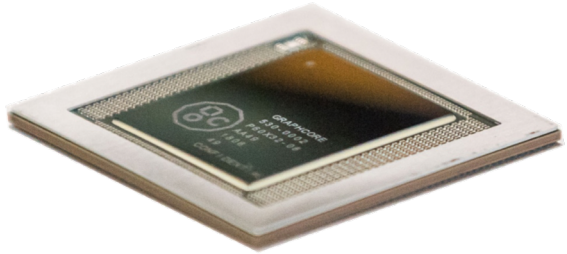
300MB IN-PROCESSOR-MEMORY™
45TB/s memory bandwidth per chip
the whole model held inside the processor

PCIe Gen4 x16
64GB/s bi-directional host
communication bandwidth



1,216 independent IPU-CORES™
each with IN-PROCESSOR-MEMORY™ tile
> 100GFLOPS per IPU-CORE™
> 7,000 programs executing in parallel

8TB/s all to all IPU-EXCHANGE™
non-blocking, any communication pattern



BRIDGING PARALLEL HARDWARE TO SOFTWARE

IPU is the world first BSP processor

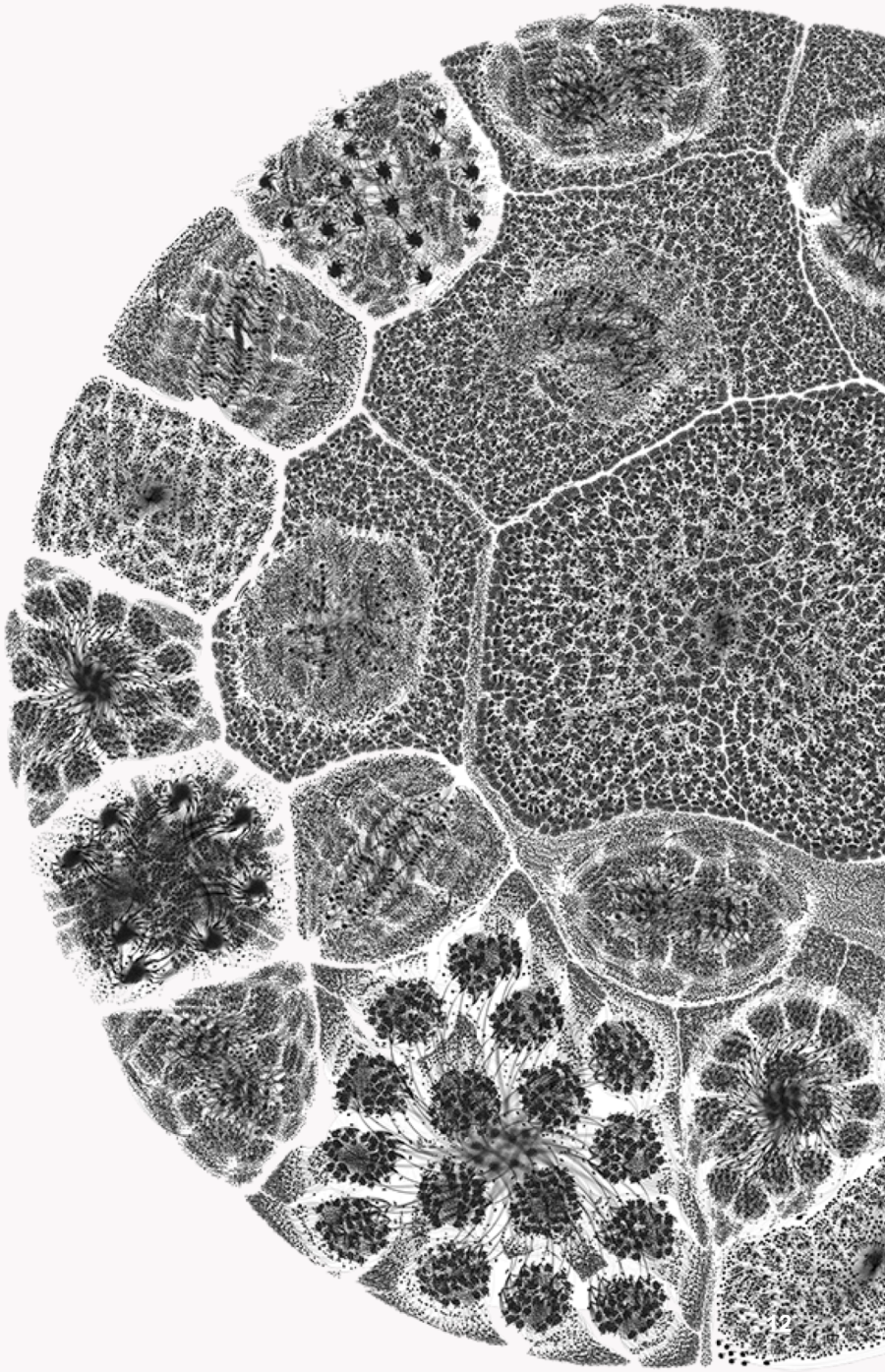
Bulk Synchronous Parallel (BSP)
compute | synchronize | exchange

Easy to program
no live locks or dead locks

Widely used in compute clusters:
Google | Facebook | ...

POPLAR®

Software stack



POPLAR®

Software stack

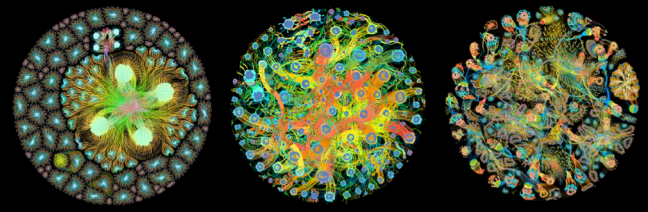
SEAMLESS INTERFACE TO INDUSTRY STANDARD ML FRAMEWORKS

OPTIMIZED GRAPH MAPPING AND CODE COMPILER BUILT USING LLVM

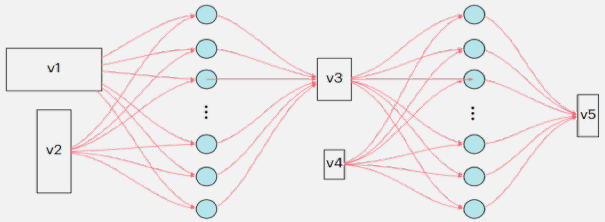
POPLAR GRAPH COMPILER

IPU CPU SIM

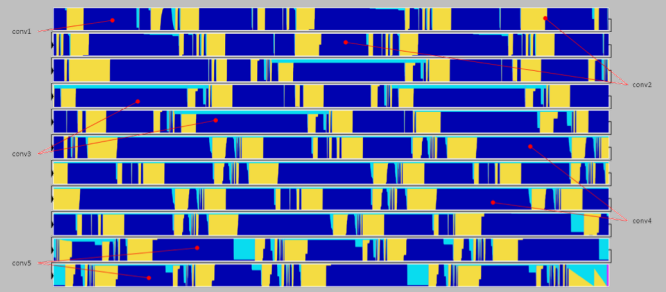
DRIVERS, UTILITIES AND GRAPH ENGINE FOR EXECUTION



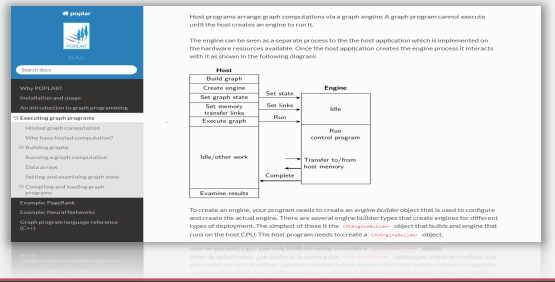
POPLAR® GRAPH PROGRAMMING FRAMEWORK (C++ & PYTHON) AND OPEN SOURCE ML LIBRARIES



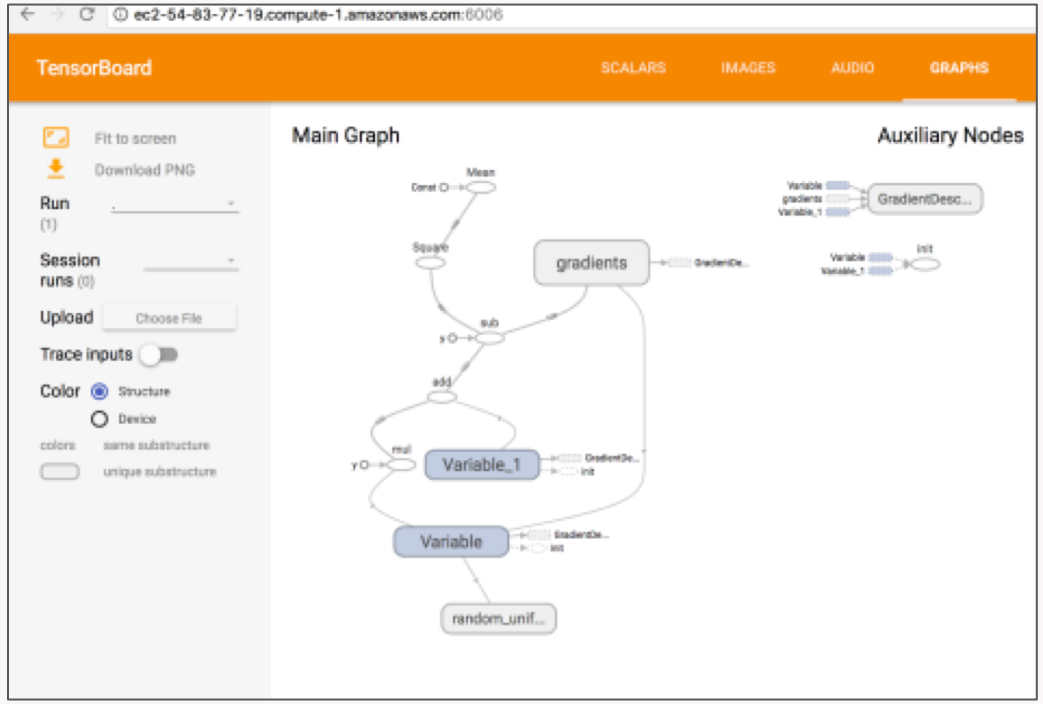
ADVANCED VISUALIZATION AND DEBUG TOOLS



COMPREHENSIVE USER DOCUMENTATION, EXAMPLES, APPLICATION NOTES AND TUTORIALS



DEVELOP YOUR MODEL USING INDUSTRY STANDARD ML FRAMEWORKS



TensorFlow

PYTORCH

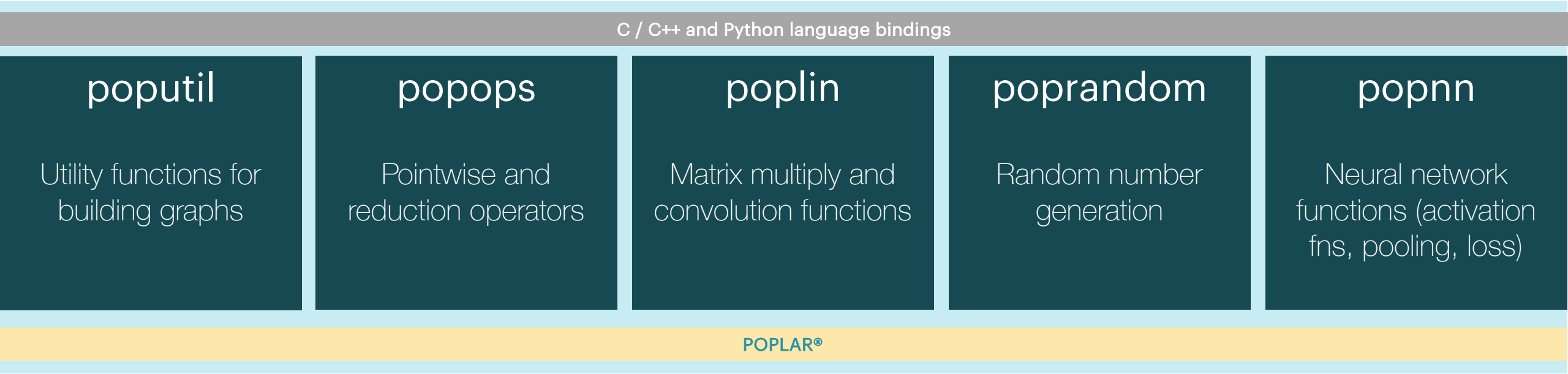
ONNX

mxnet

Optimized support for inference and training

POPLIBS™

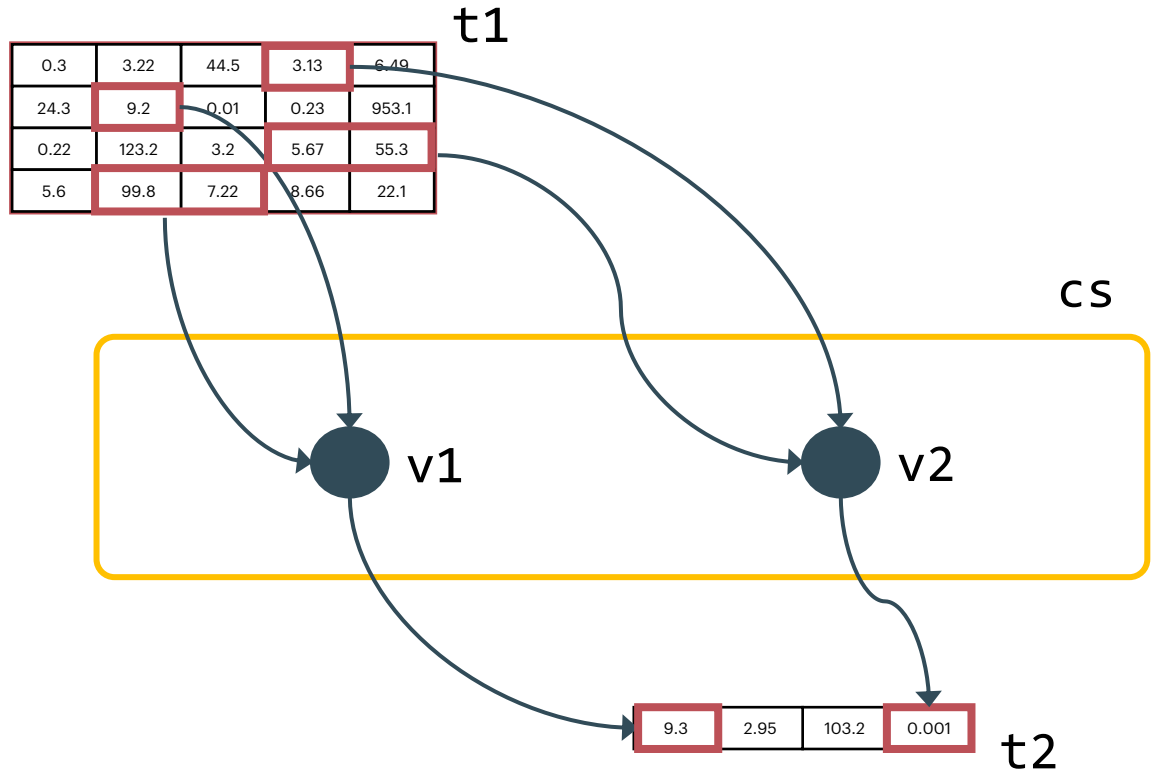
Highly optimized ***open source*** libraries partition work and data efficiently across IPU devices



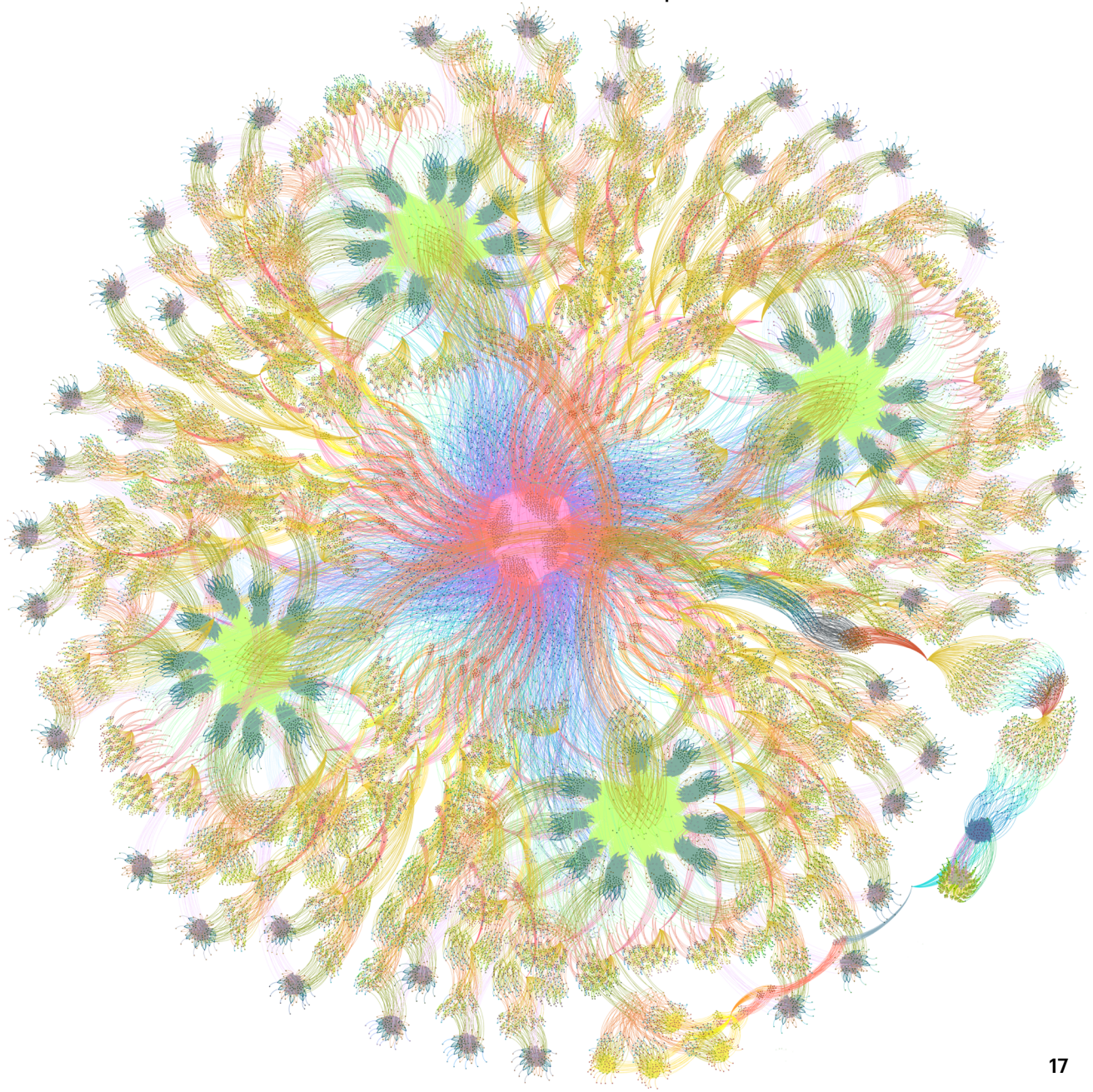
GitHub
github.com/graphcore/poplibs

POPLAR® C++ / PYTHON, GRAPH FRAMEWORK LETS YOU MODIFY OR CREATE YOUR OWN LIBRARY ELEMENTS

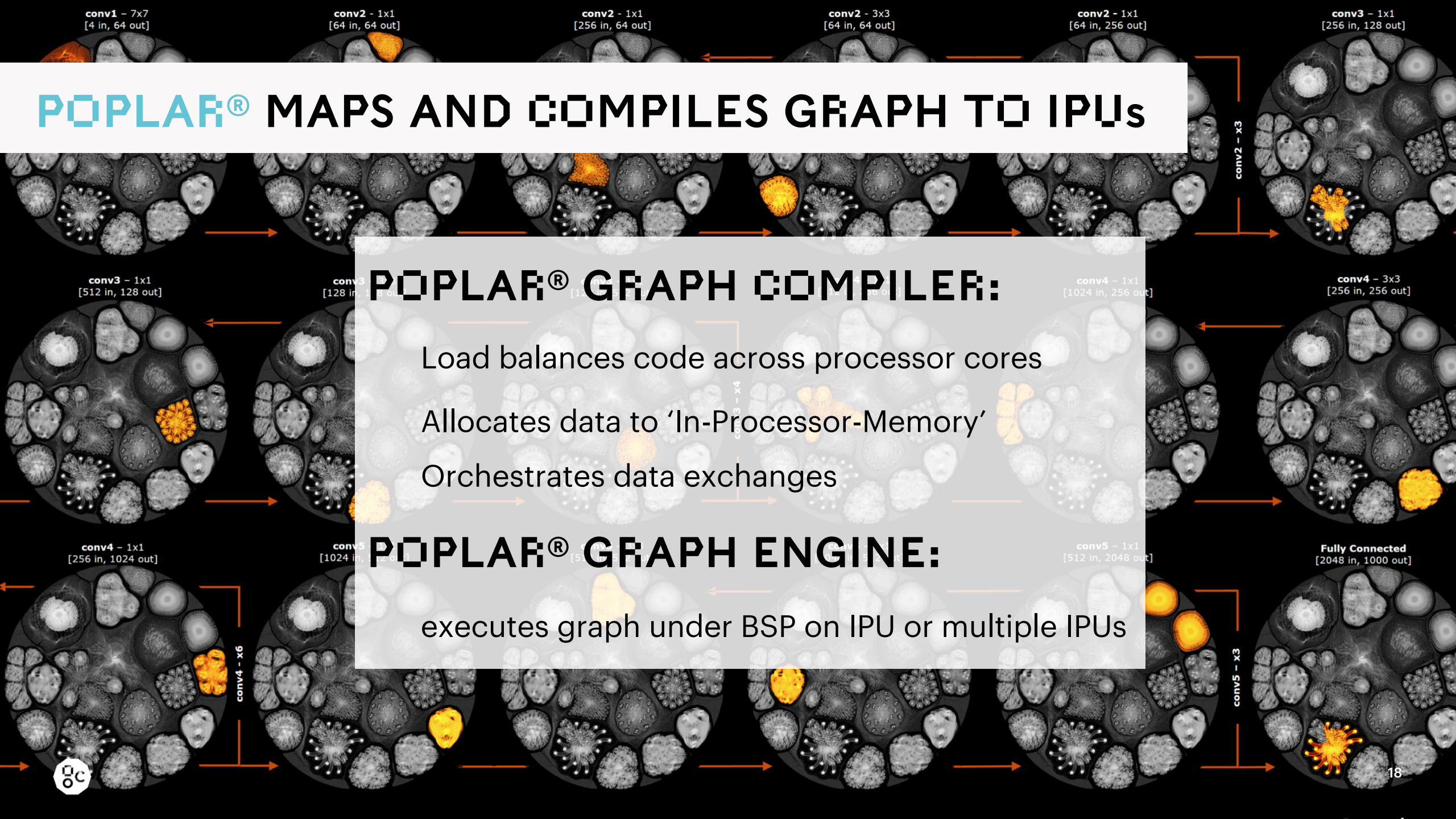
```
Graph g(device);
g.addCodelets("codelets.cpp");

Tensor t1 = g.addTensor(“float”, {4, 5});
Tensor t2 = g.addTensor(“float”, {4});

ComputeSet cs = g.addComputeSet(“myComputeSet”)

VertexRef v1 = g.addVertex(cs, “AdderVertex”);
VertexRef v2 = g.addVertex(cs, “AdderVertex”);

g.connect(t1[1][1], v1["x"]);
g.connect(t1.slice({3, 1}, {4, 3}), v1["y"]);

g.connect(t2[0], v1["z"]);

g.connect(t1[0][3], v2["x"]);
g.connect(t1.slice({2, 2}, {3, 4}), v2["y"]);
g.connect(t2[3], v2["z"]);

g.setTileMapping(t1.slice({0, 0}, {4, 2}), 0);
g.setTileMapping(t1.slice({0, 2}, {4, 5}), 1);
g.setTileMapping(t2, 2);

g.setTileMapping(v1, 0);
g.setTileMapping(v2, 1);
```



POPLIBS™ and **POPLAR®** expand ML Framework output to a full compute graph.



DeepVoice GRAPH VISUALIZATION FROM POPLAR®



POPLAR® MAPS AND COMPILES GRAPH TO IPUs

POPLAR® GRAPH COMPILER:

Load balances code across processor cores

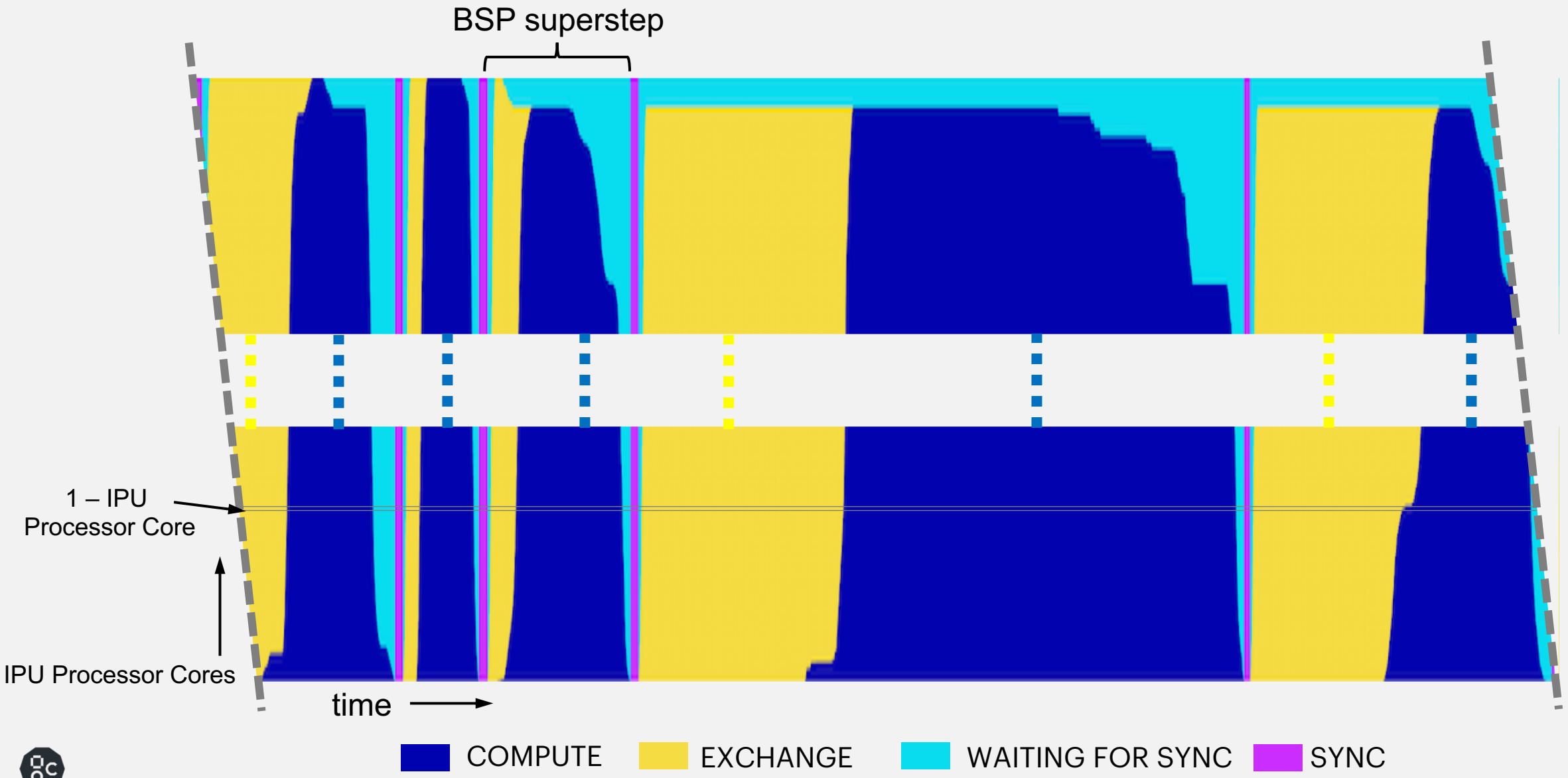
Allocates data to 'In-Processor-Memory'

Orchestrates data exchanges

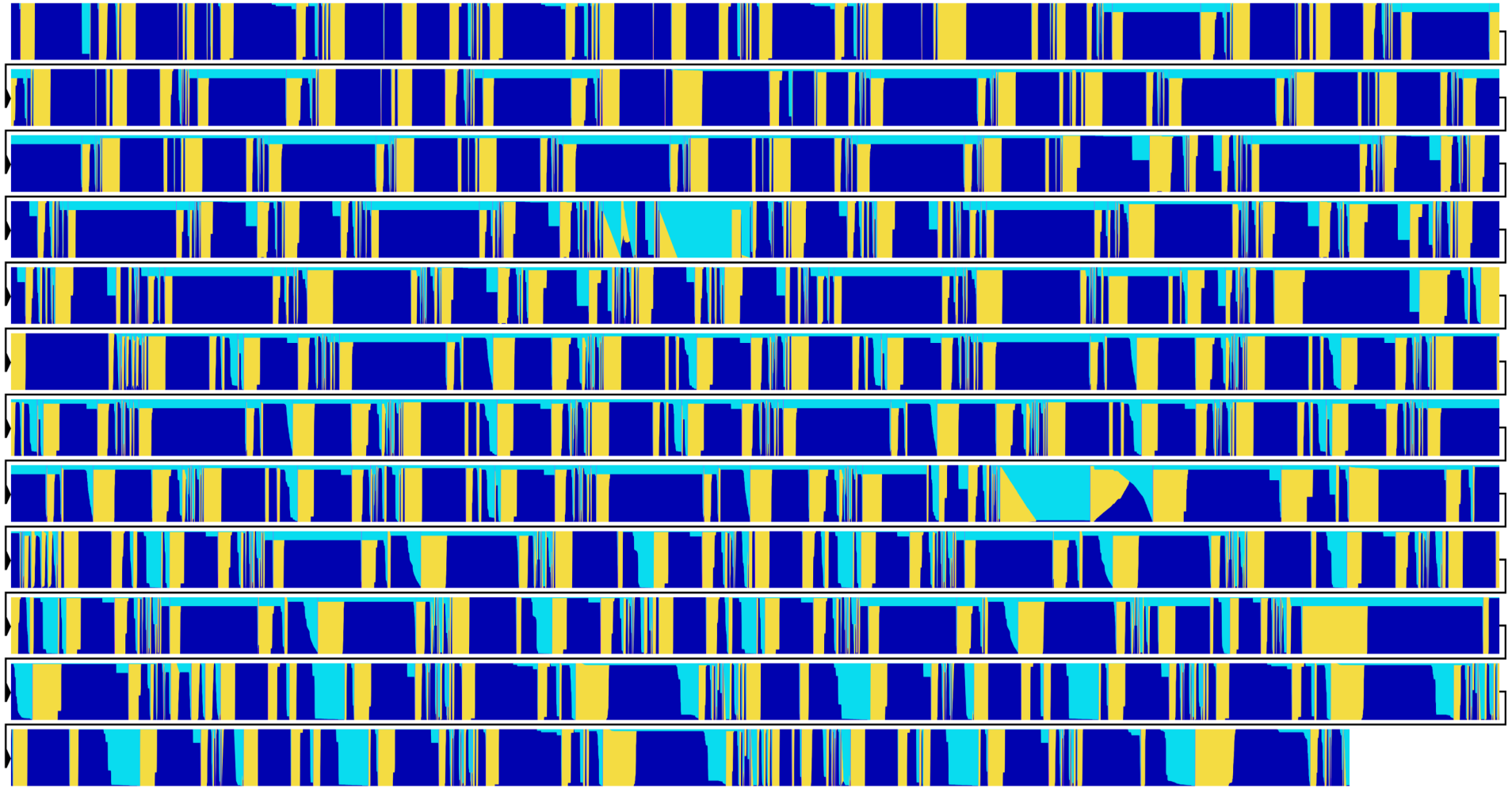
POPLAR® GRAPH ENGINE:

executes graph under BSP on IPU or multiple IPUs

ADVANCED VISUALIZATION AND DEBUG TOOLS



EXAMPLE BSP TRACE : RESNET-50 TRAINING

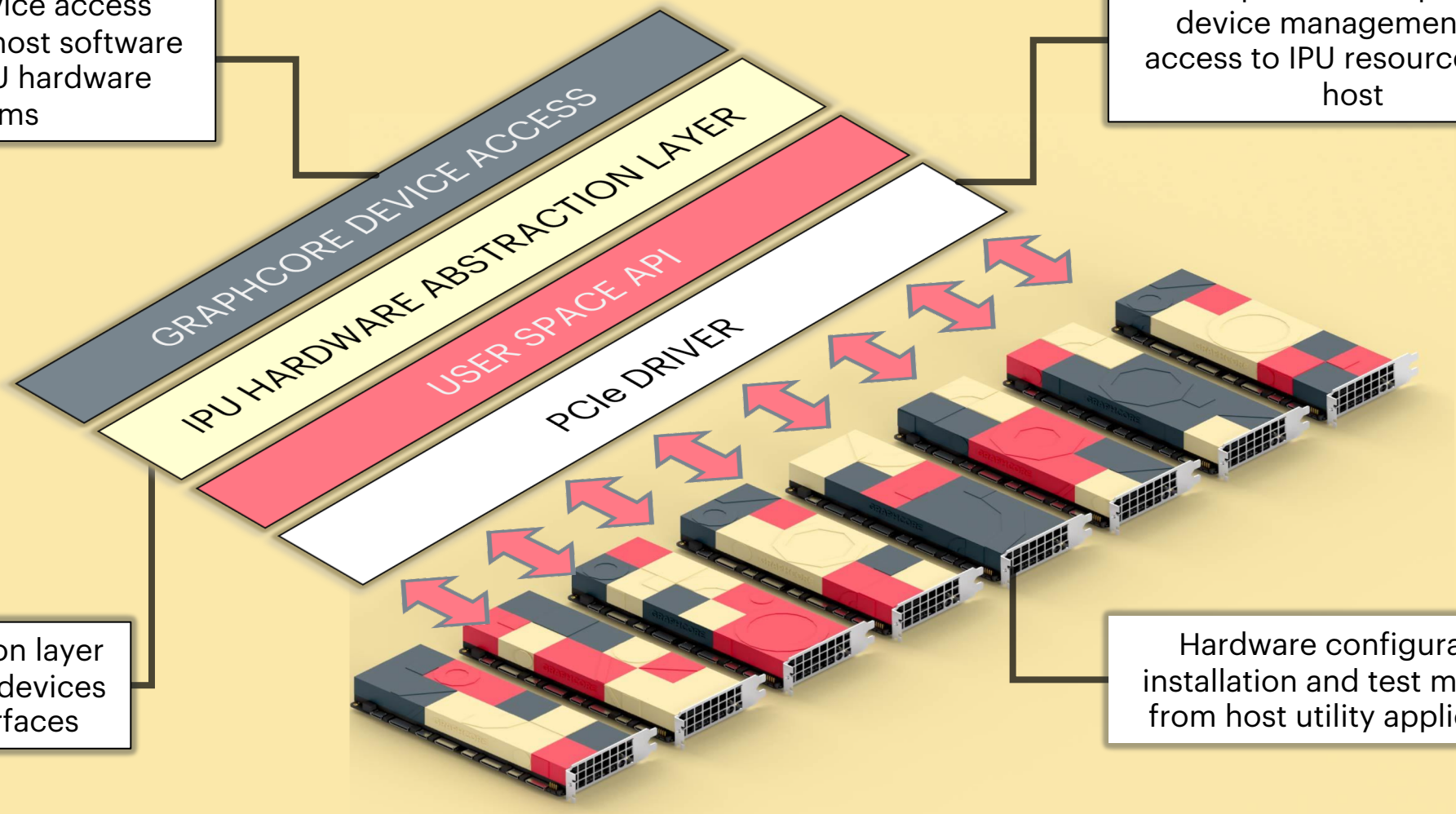


COMPUTE EXCHANGE WAITING FOR SYNC SYNC

DRIVERS AND UTILITIES

Graphcore device access library provides host software interface to IPU hardware platforms

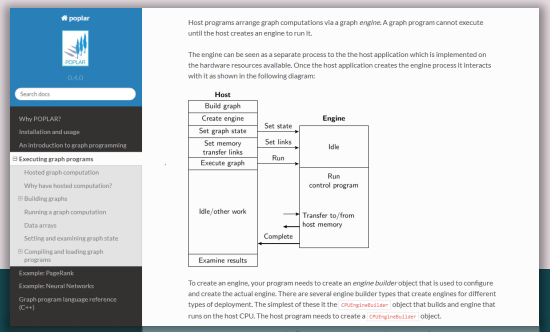
PCI Express driver providing device management and access to IPU resources from host

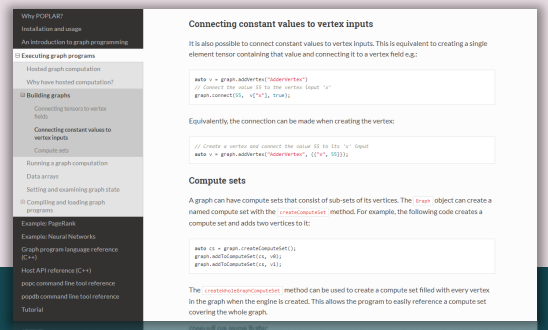


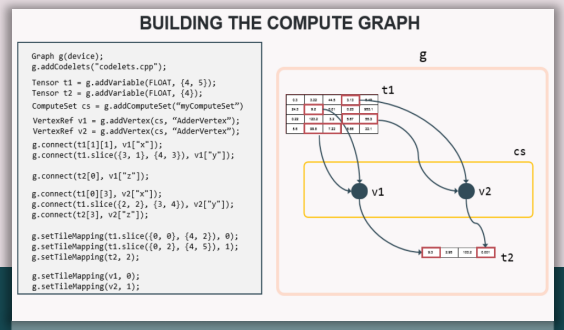
IPU hardware abstraction layer allows provision of IPU devices via multiple host interfaces

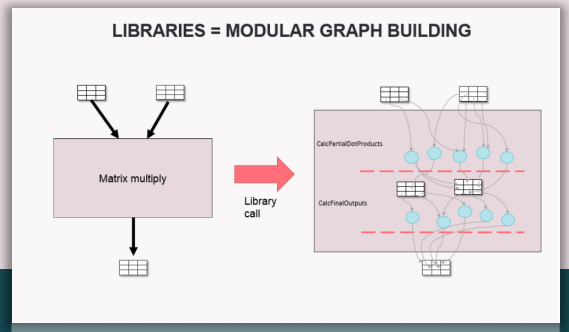
Hardware configuration, installation and test managed from host utility applications

COMPREHENSIVE DOCUMENTATION







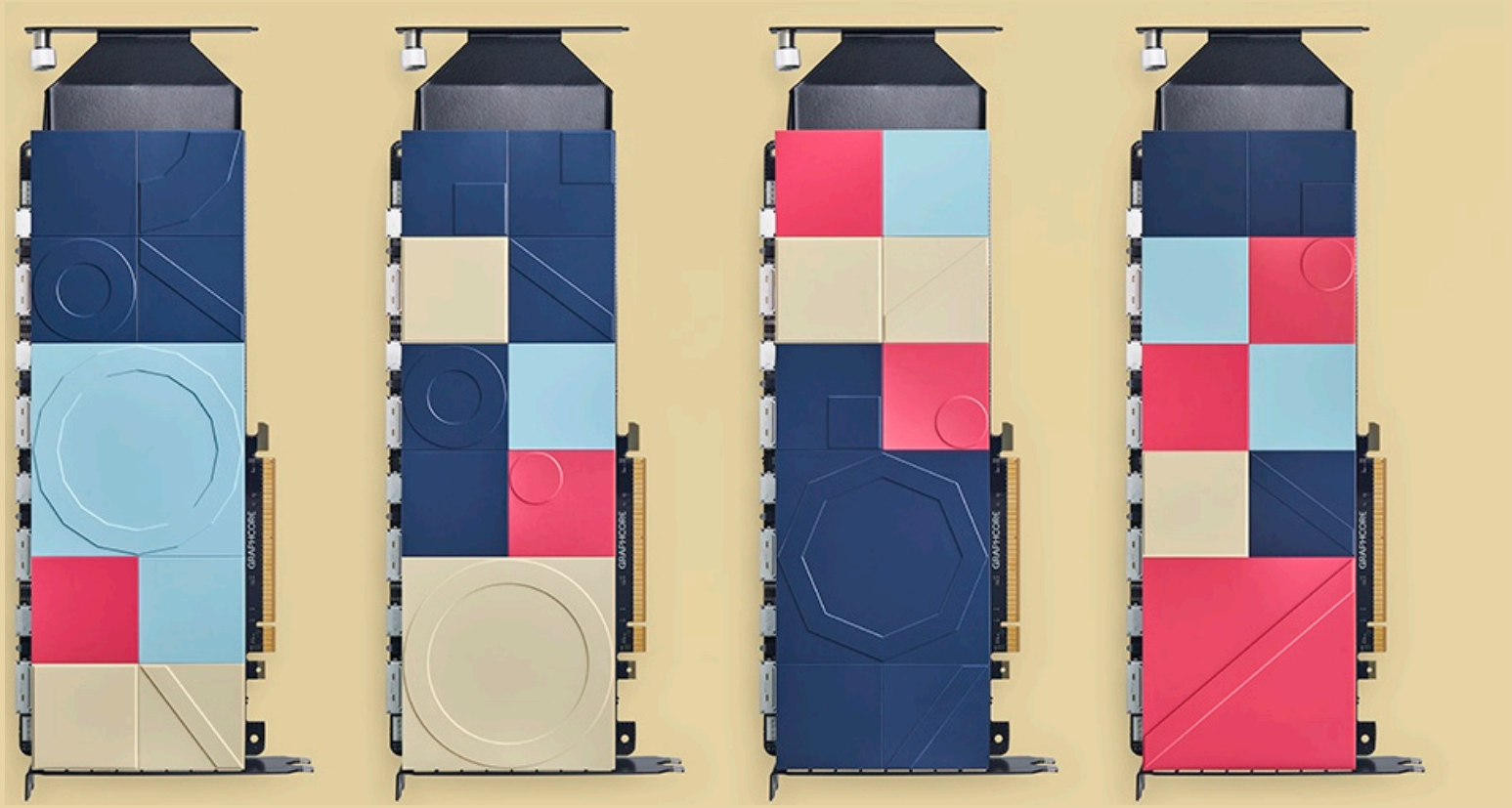


DOCUMENTATION, CODELET EXAMPLES, APPLICATION NOTES, AND TUTORIALS including VIDEOS

PRODUCTS

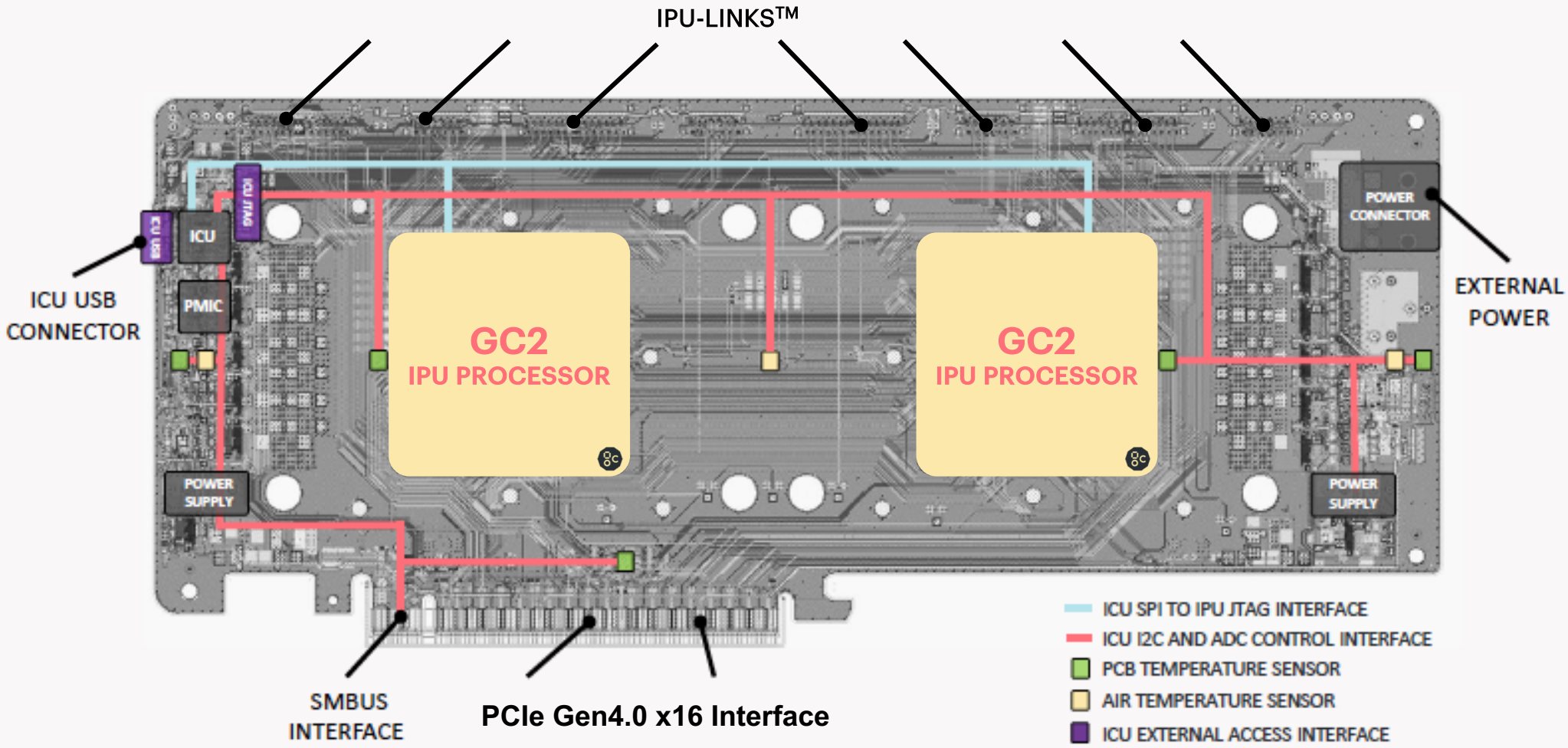
C2 IPU-Processor PCIe Card

C2 IPU-PROCESSOR PCIe CARD



DOUBLE WIDTH PCIE CARD WITH 2 – COLOSSUS **GC2** IPU PROCESSORS
CARD-TO-CARD **IPU-LINKS™** *(320GB/s)*
250 TERA-FLOP MIXED PRECISION IPU COMPUTE @ 300W

C2 IPU-PROCESSOR PCIe CARD



ResNet50 network visualization from POPLAR™

WE HAVE DEVELOPED A NEW KIND OF HARDWARE THAT WILL LET INNOVATORS CREATE THE NEXT GENERATION OF MACHINE INTELLIGENCE

THANK YOU