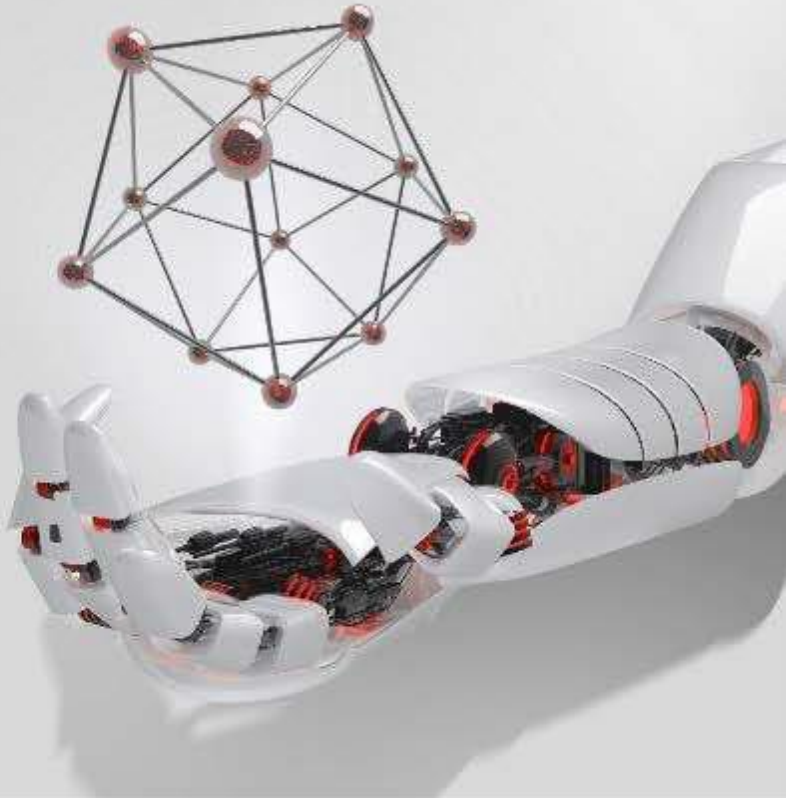
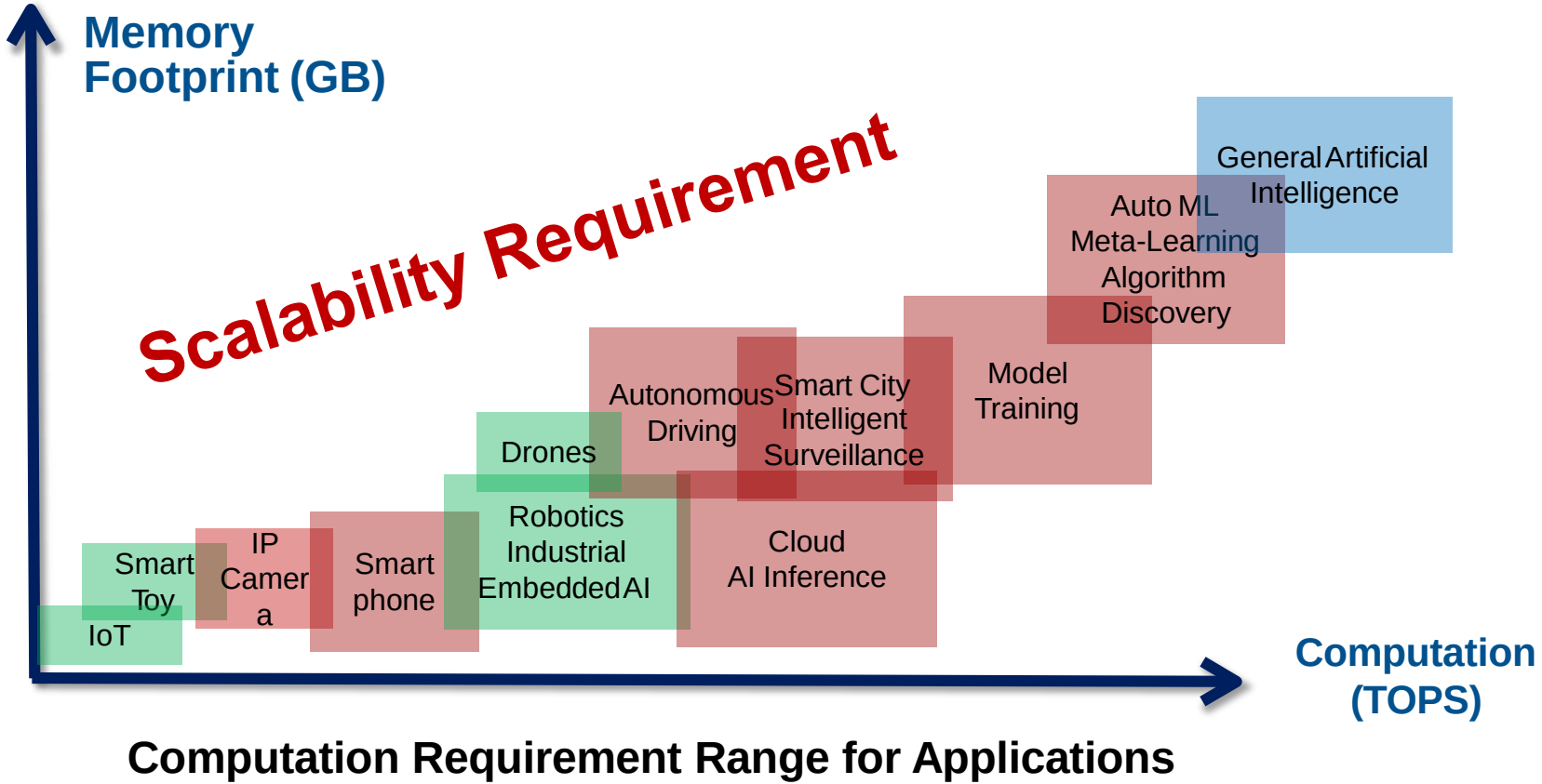
DaVinci: A Scalable Architecture for Neural Network Computing



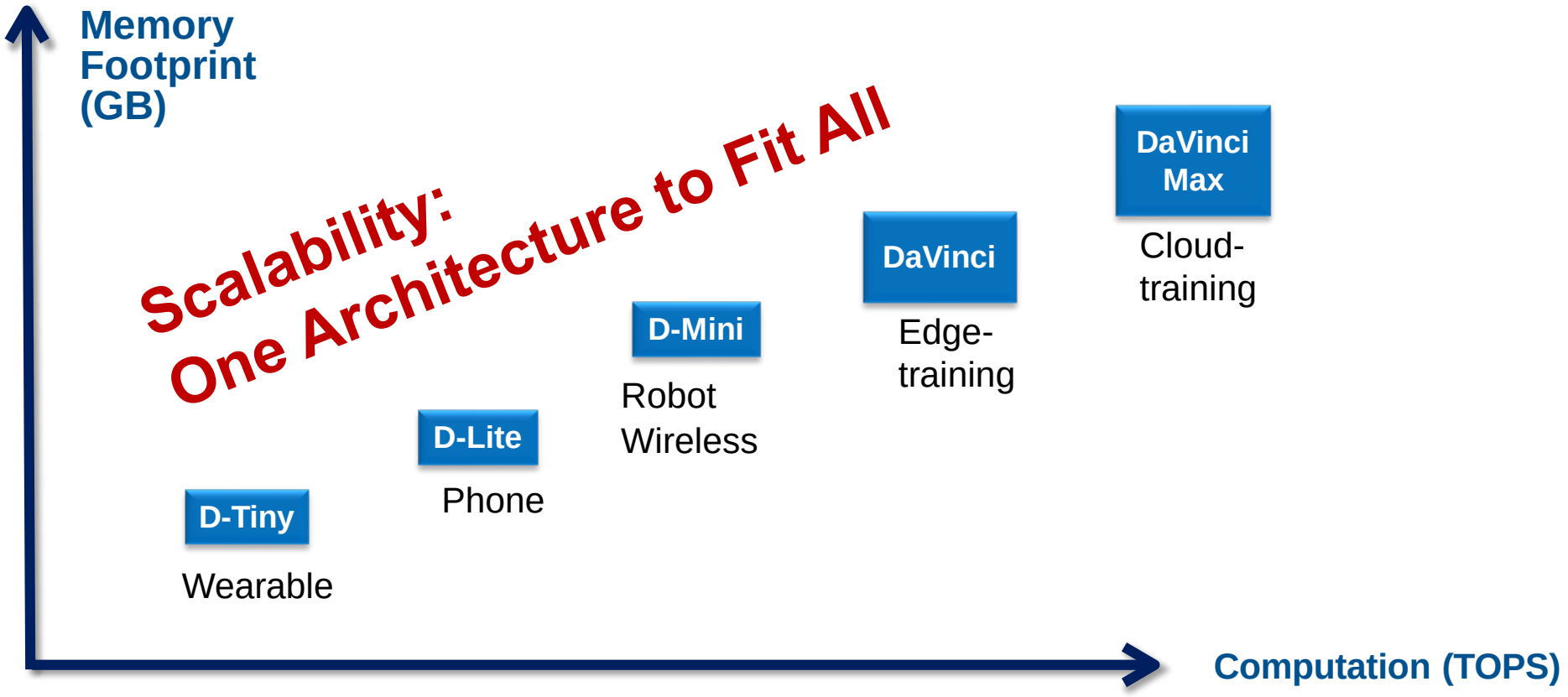
HUAWEI

AI Computation for All Scenarios

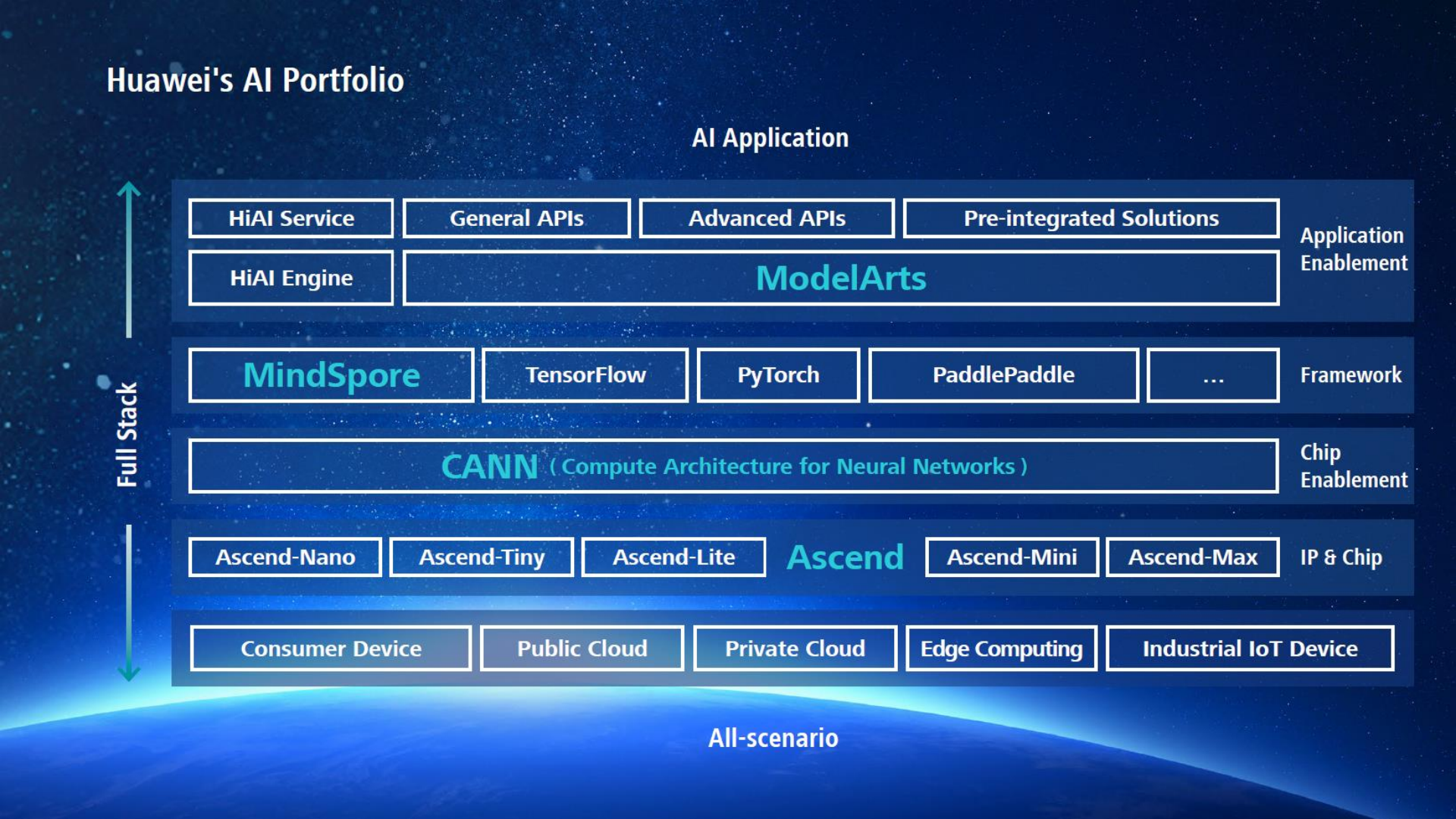


Computation Requirement Range for Applications

Huawei AI Processor for All Scenarios



Huawei's AI Portfolio

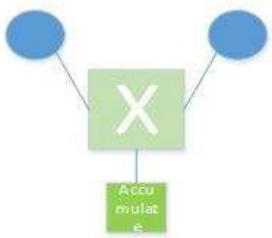


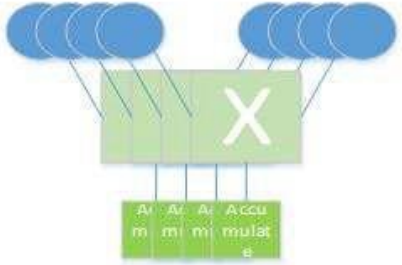
Architecture Overview of DaVinci

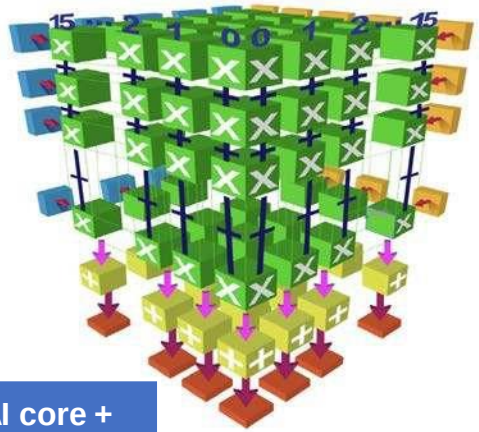
Building Blocks and their Computation Intensity

1D Scalar Unit + **2D Vector Unit** + **3D Matrix Unit**

Full flexibility | Rich & efficient operations | High intensity



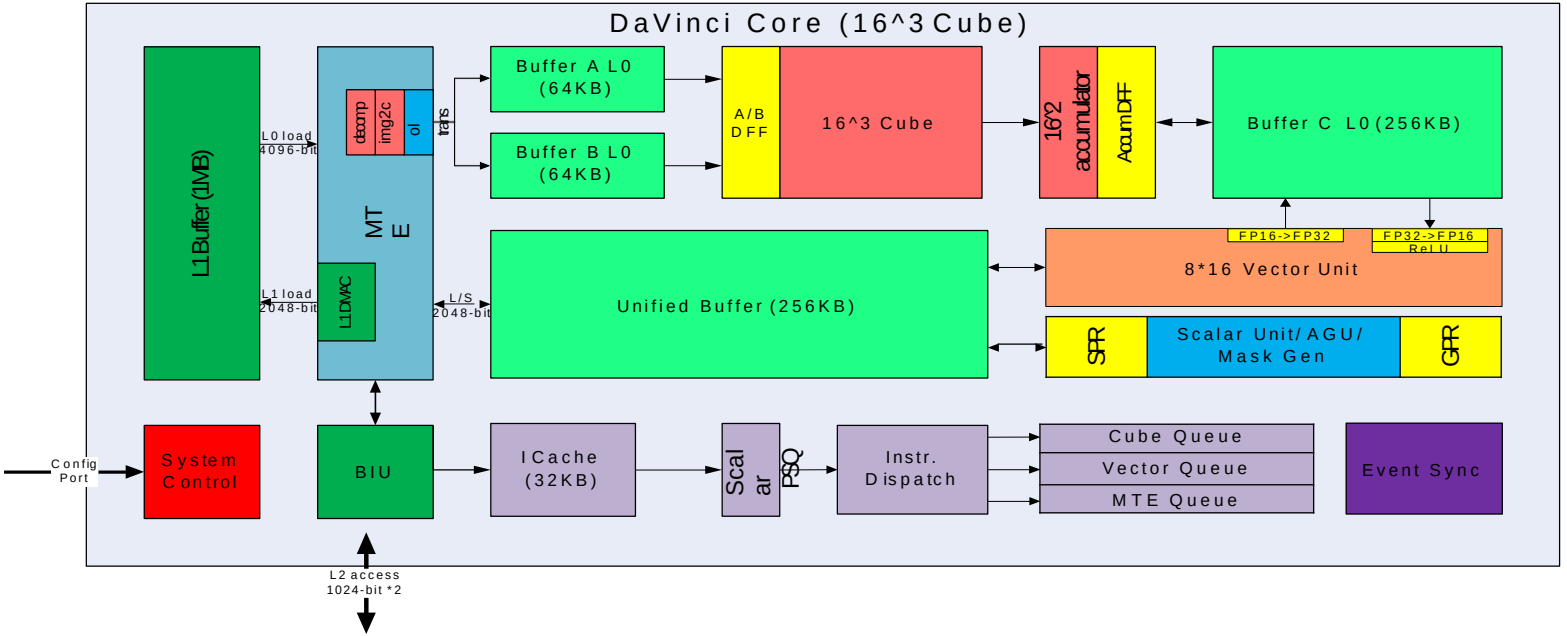




N	N^2	N^3
1	1	1
2	4	8
4	16	128
8	64	512
16	**256**	**4096**
32	1024	32768
64	4096	262144

	GPU + Tensor core	AI core + SRAM
Area (normalized to 12 nm)	5.2mm^2	13.2mm^2
Compute power	1.7Tops fp16	8Tops fp16

DaVinci Core

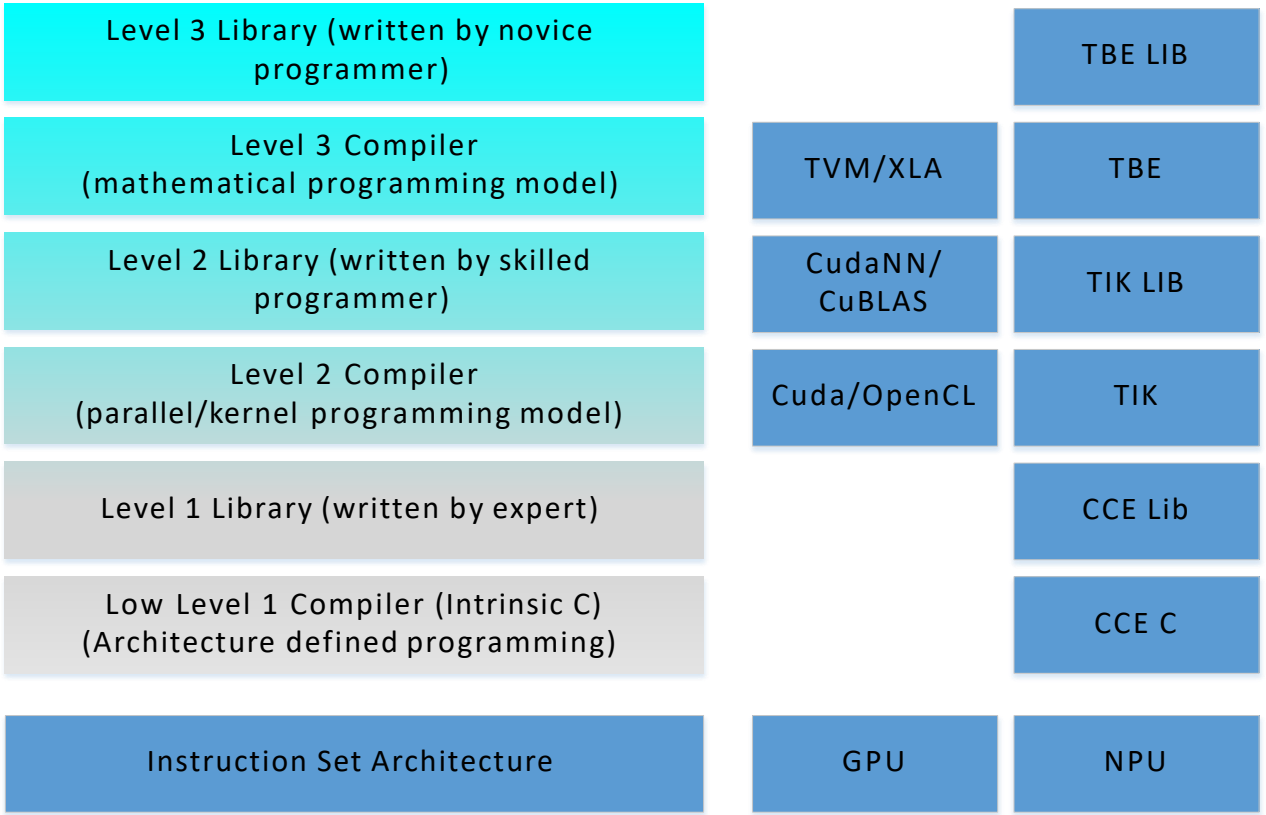


- **Cube**: 4096(16^3) FP16 MACs + 8192 INT8 MACs
- **Vector**: 2048bit INT8/FP16/FP32 vector with special functions
  (activation functions, NMS- Non Minimum Suppression, ROI, SORT)
- Explicit memory hierarchy design, managed by MTE

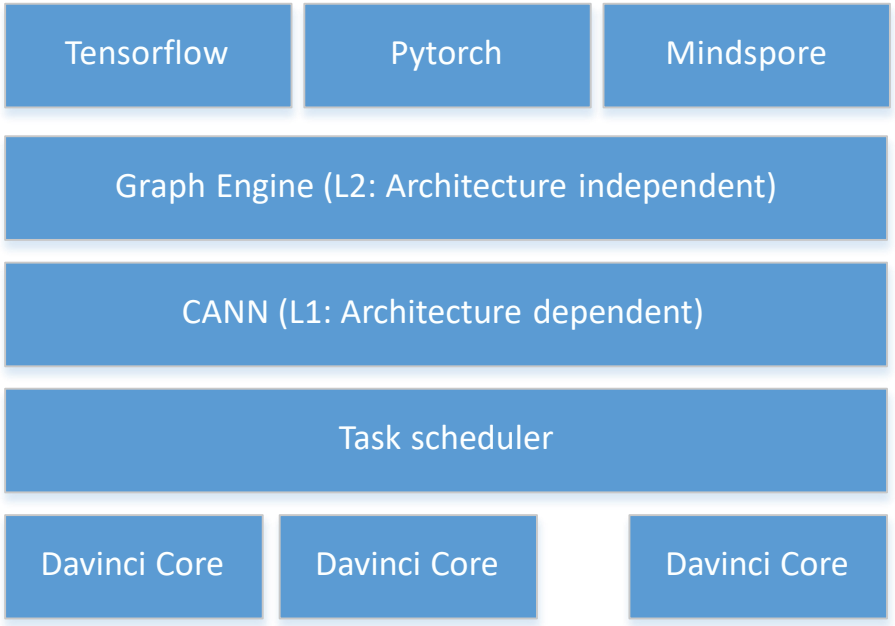
Micro Architecture Configurations

Core Version	Cube Ops/cycle	Vector Ops/Cycle	L0 Bus width	L1 Bus Width	L2 Bandwidth
Davinci Max	8192	256	Match Execution Units Not bottleneck	A:8192 B:2048	910: 3TB/s ÷32 610: 2TB/s ÷8 310: 192GB/s÷2
Davinci Lite	4096	128		A:8192 B:2048	38.4GB/s
Davinci Tiny	512	32		A:2048 B:512	None
	Set the performance baseline	Minimize vector bound		Ensure this is not a bound	Scarce, limited by NoC, avoid bound where possible

Overview of the Software Stack



Putting All This Together

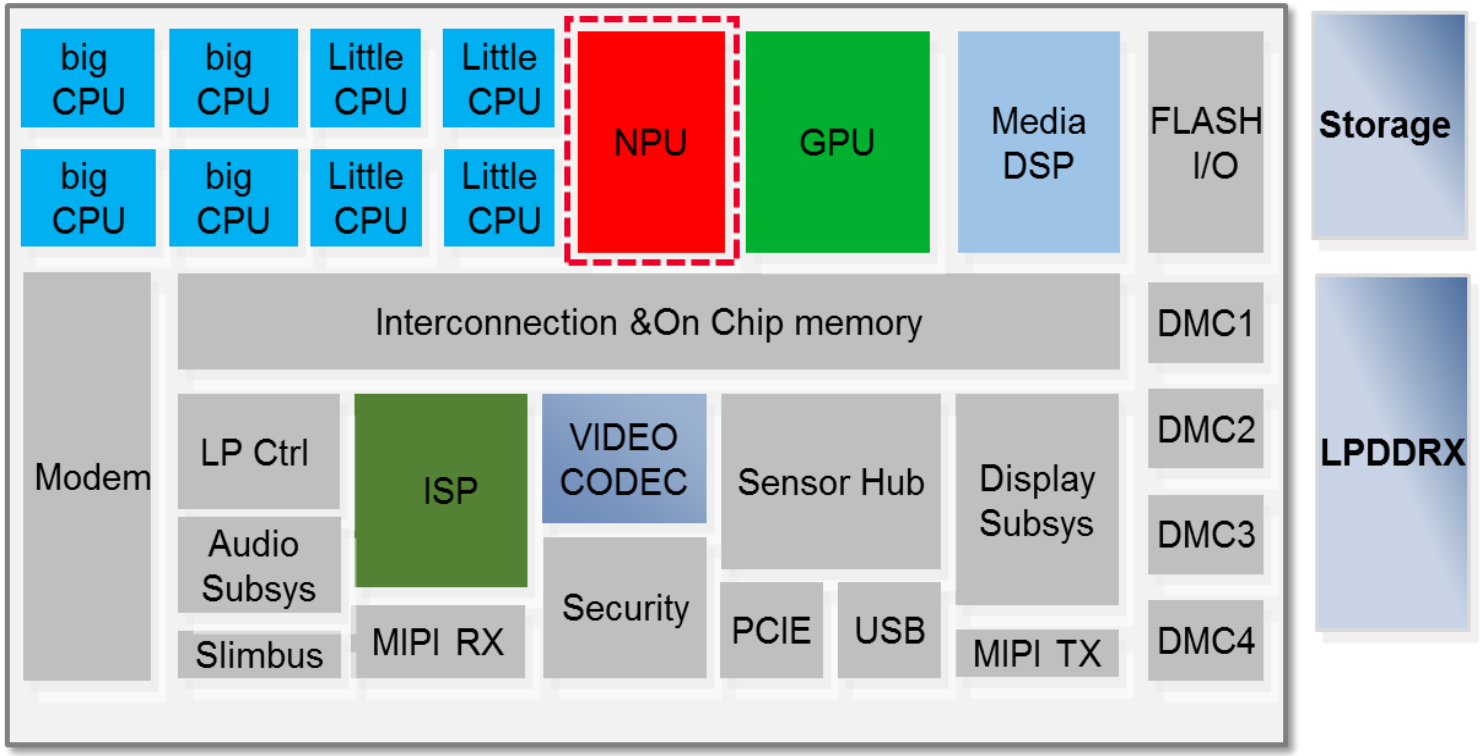


- User program AI model using familiar frameworks
- Extends operator library when necessary
- The tasks are executed in a single node, or over a network cluster

DaVinci AI Processors and Products

Mobile AP SoC: Kirin 990 contain D-lite version NPU

Mobile AP SoC: Kirin 990 contain D-lite version NPU



Ascend AI Processor： 310 and 910

Ascend 310

High Power Efficiency

- Ascend-Mini
  Architecture: DaVinci
- FP16： 8 TeraFLOPS
- INT8 ： 16 TeraOPS
- 16 Channel Video Decode – H.264/265
- 1 Channel Video Encode – H.264/265
- Power： 8W
- Process: 12nm

Ascend 910

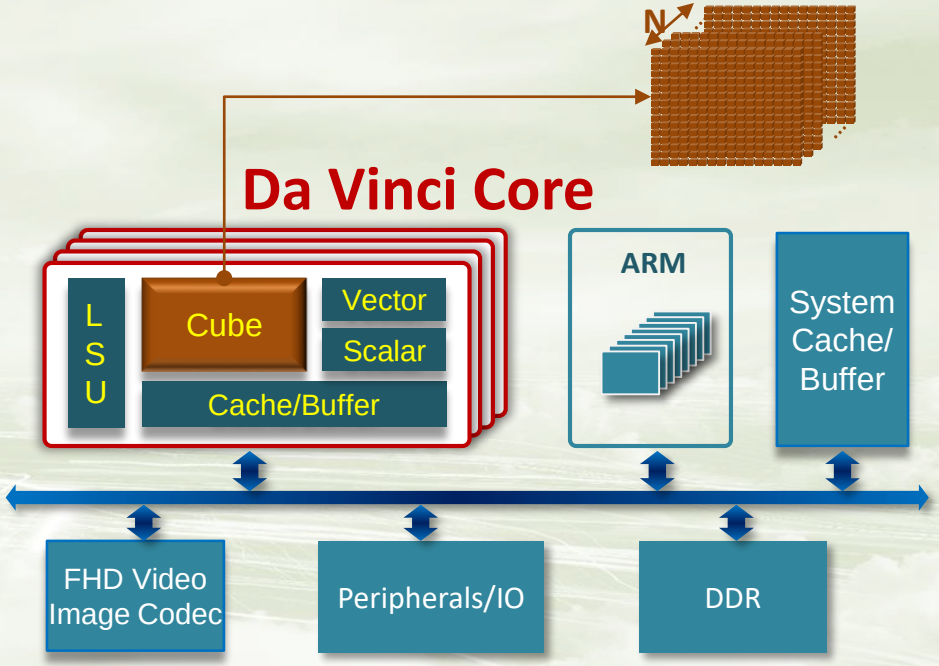
High Computing Density

- Ascend-Max
  Architecture: DaVinci
- FP16: 256 TeraFLOPS
- INT8: 512 TeraOPS
- 128 Channel Video Decode – H.264/265
- Power: 300W
- Process: 7+ nm EUV

Huawei AI Solutions For Inference

Ascend 310 Specification: D-mini version

SPECIFICATIONS	Description
Architecture	AI co-processor
Performance	Up to 8T @FP16
	Up to 16T@INT8
Codec	16 Channel Decoder – H.264/265 1080P30 1 Channel Encoder
Memory Controller	LPDDR4X
Memory Bandwidth	2*64bit @3733MT/S
System Interface	PCIe3.0 /USB 3.0/GE
Package	15mm*15mm
Max Power	8Tops@4W, 16Tops@8W
Process	12nm FFC

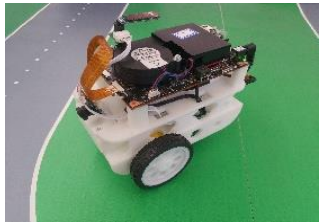


Note: This is typical configuration, high performance and low power sku can be offered based on your requirement.

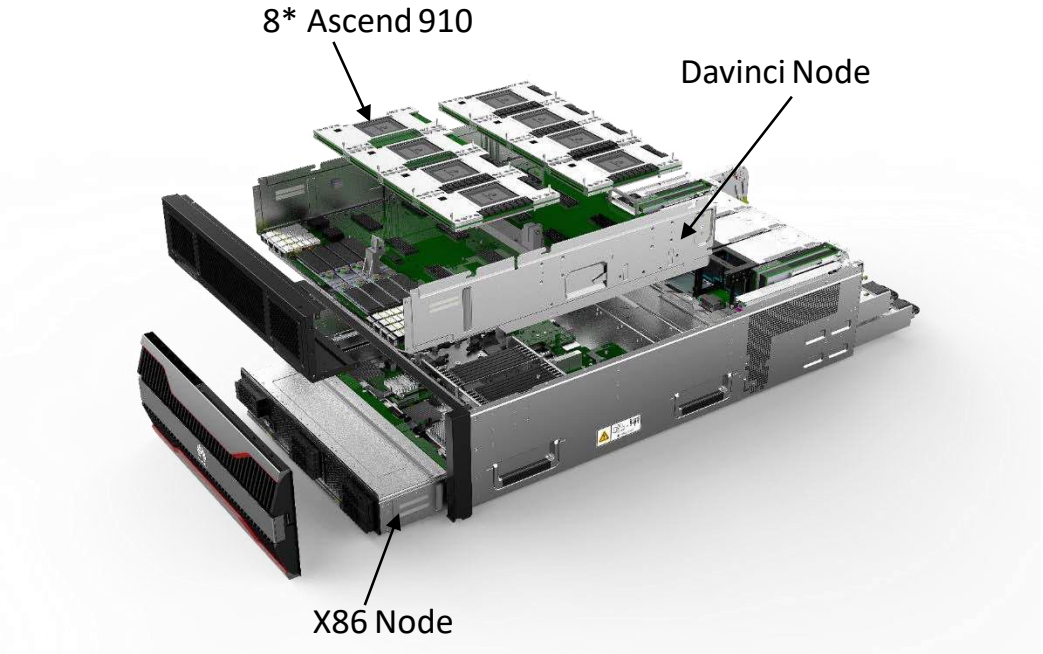
HUAWEI

Applications Built on Atlas 200 DK

Semantic assisted high-precision 3D reconstruction
Image edge detection
Facial recognition
Vehicle detection
Facial expression transplant
Blocked face detection
Image marking
Facial recognition
Unmanned vehicle
Handwritten text recognition
Robots
HDR
Image processing
Robotic arm
AR shadow generation
Age recognition
Fundus retinal vascular segmentation
Classification network
Cartoon image generation
NPL
Protein subcellular position prediction

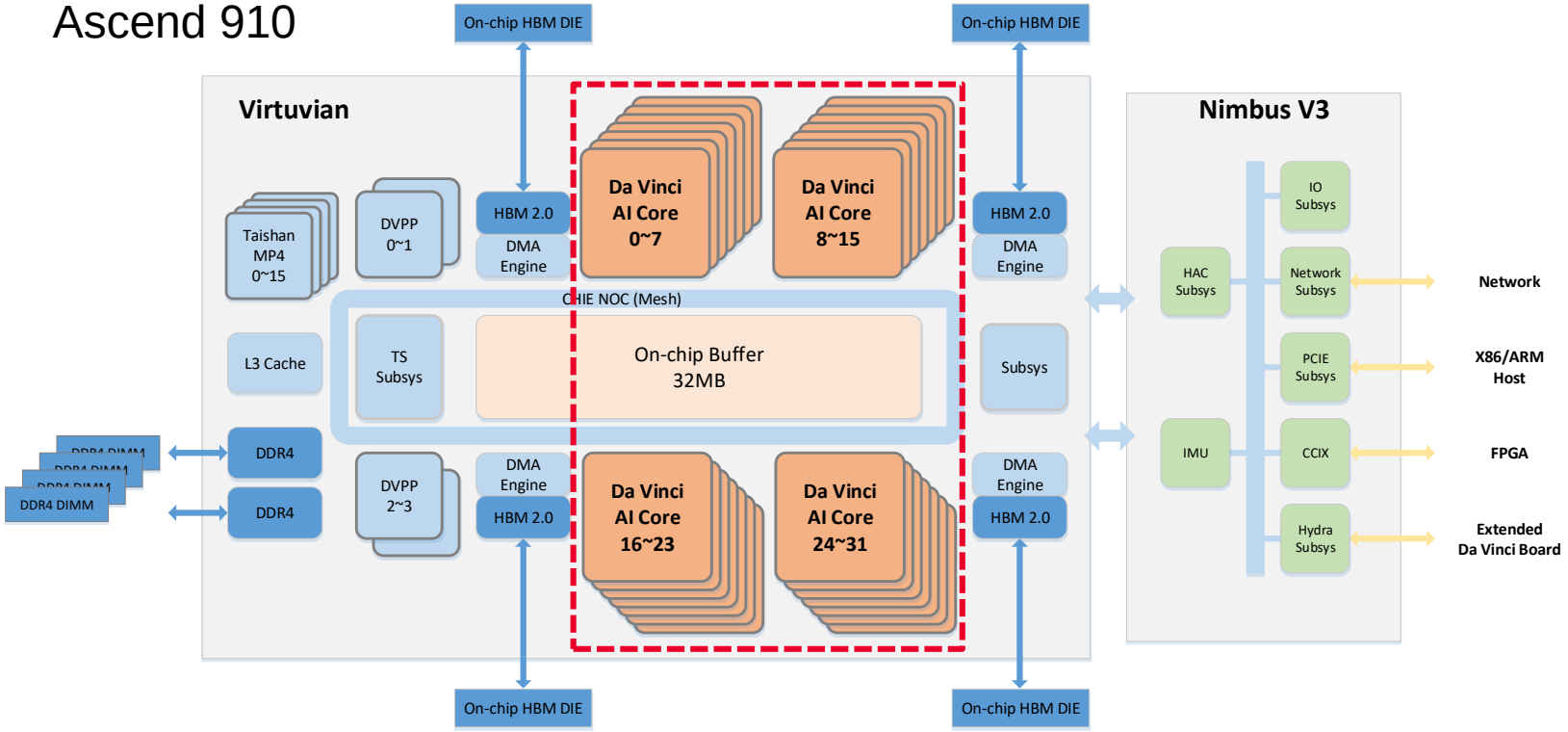


Huawei AI Solutions For Training: Ascend 910 Server

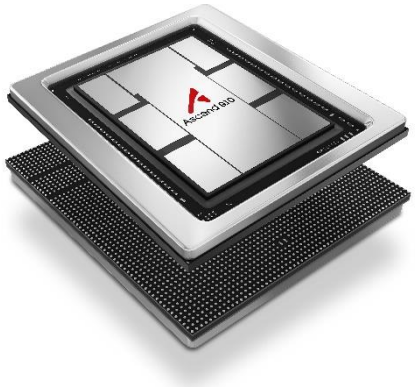


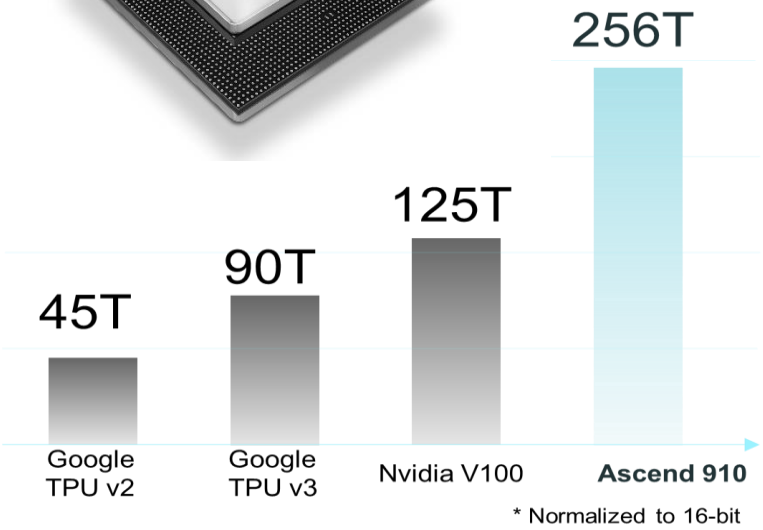
AI Training SoC: Atlas 910 contains D-max version

Ascend 910



Ascend 910 Specification

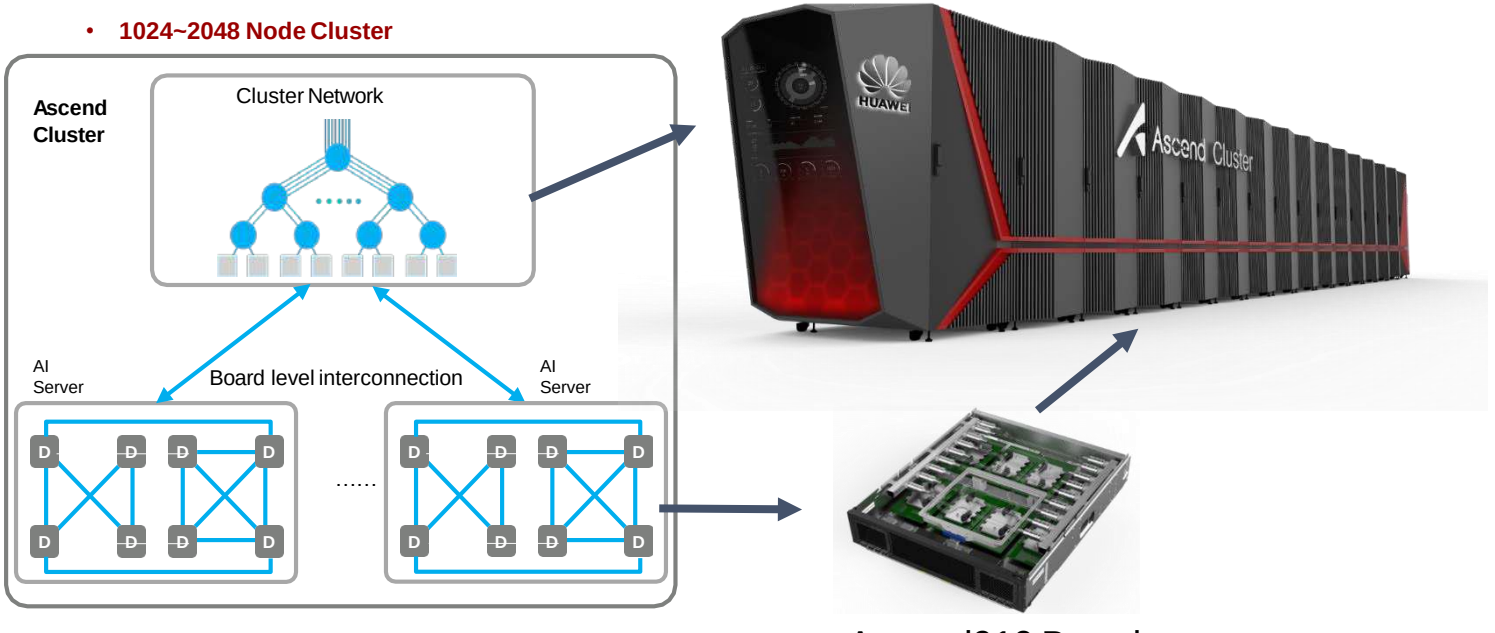




SPECIFICATIONS	Description
Architecture	AI co-processor
Performance	Up to 256T @FP16
	Up to 512T@INT8
Decoder Codec	128 Channel FHD Video Decoder – H.264/265
Memory Interface	32GB HBM Gen2
System Interface	PCIe3.0 x16 & HCCS & 100G RoCE
Max Power	Up to 350W
Process	7nm+

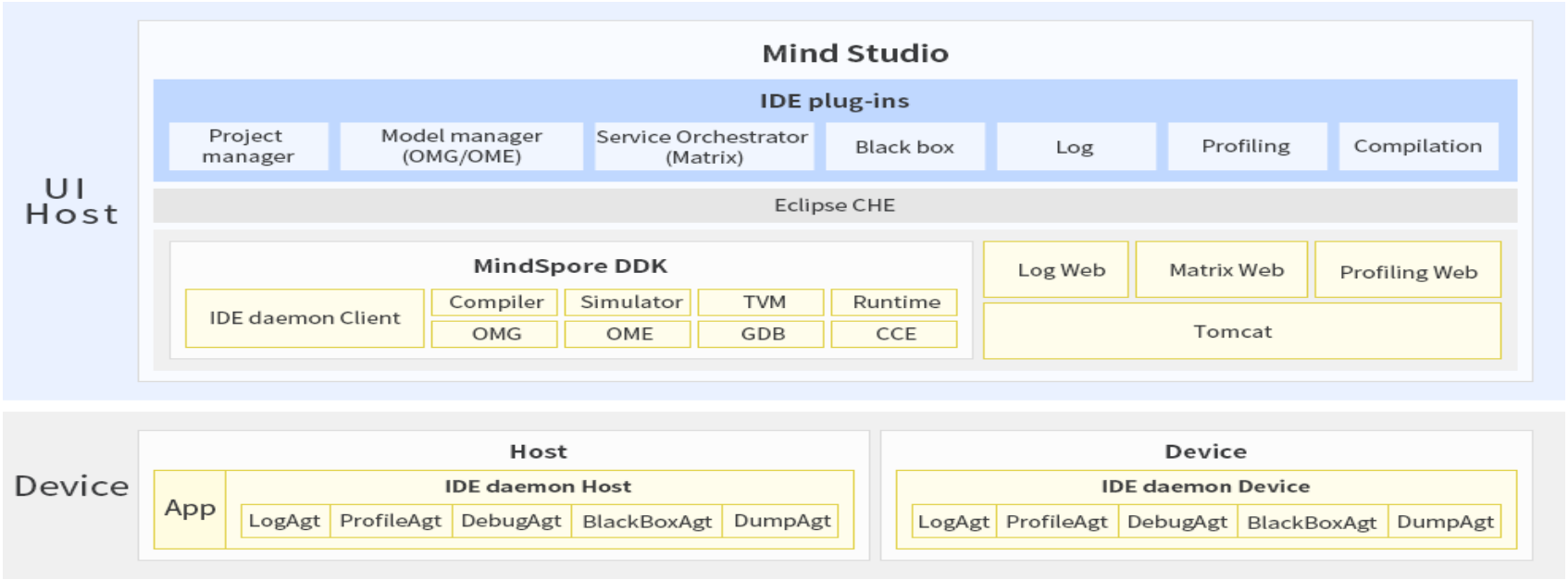
Ascend 910 Cluster

- 2048 Node x 256TFlops = 512 Peta Flops



Development Tool: Mind Studio

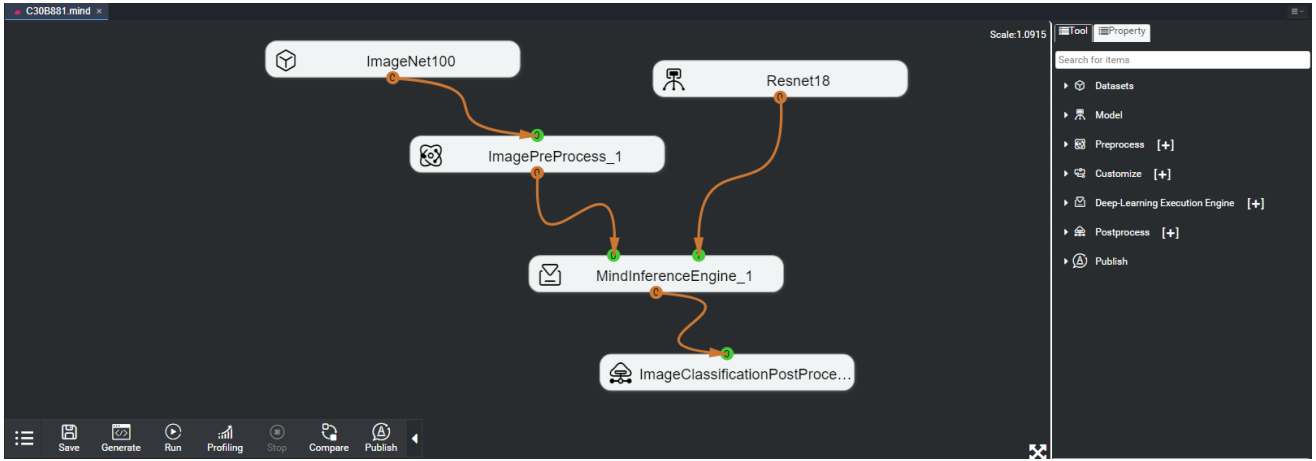
Mind Studio —— Atlas 200 DK Tool Chain



- **Operator development**
- **Offline model tool**
- **Service orchestration tool**
- **App Development Tool**

Service Orchestrator

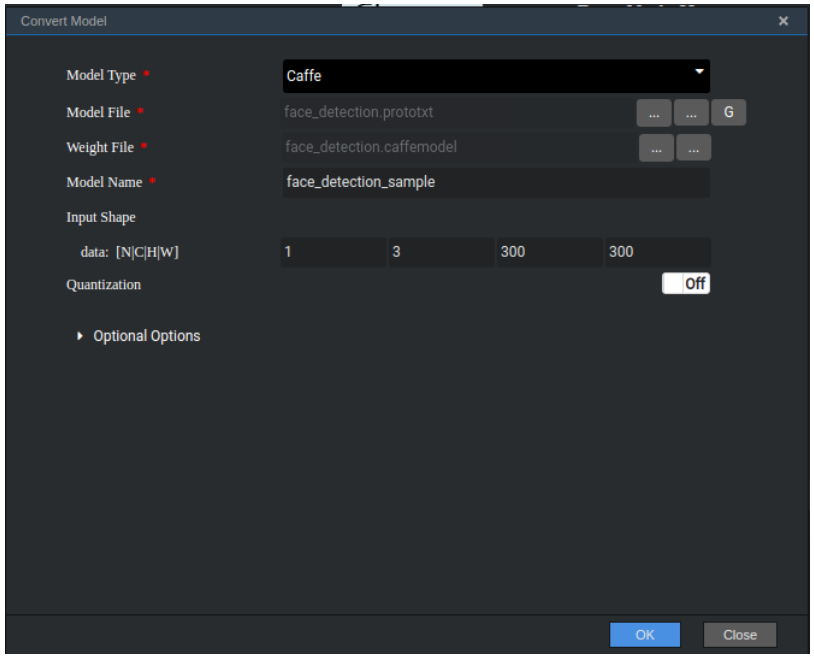
Drag-and-drop mode: auto-generate codes

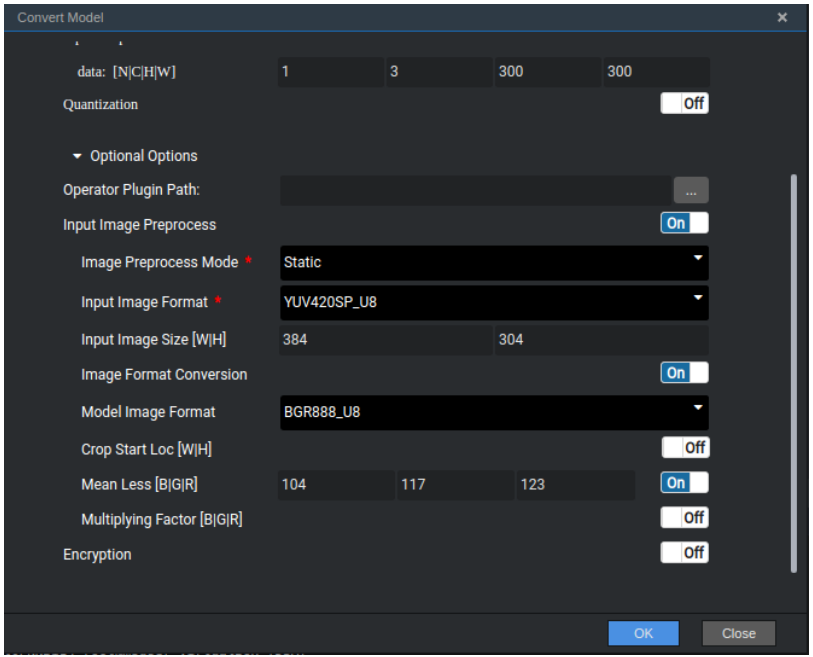


Model Manager

Wizard for OMG, configuring AI preprocessing (AIPP) inside of offline model.

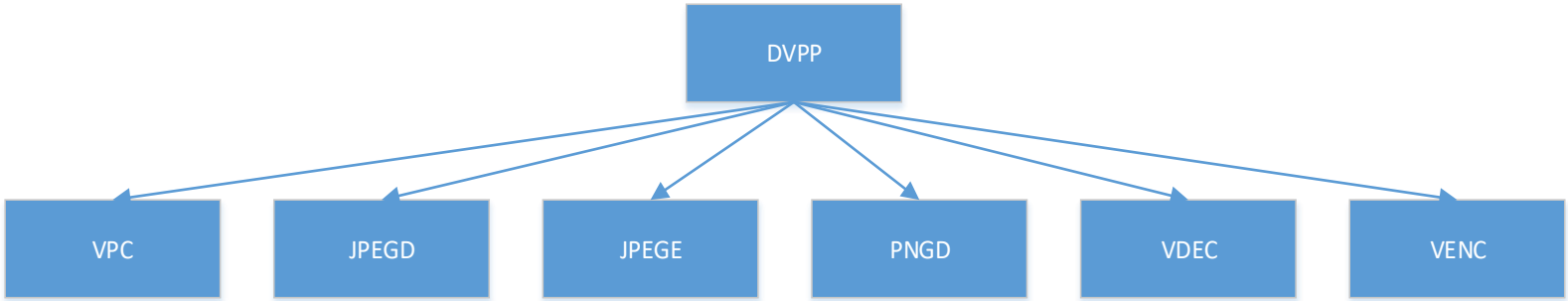
AIPP involves image cropping, color space conversion (CSC), and mean subtraction and multiplication coefficient (pixel changing). All these functions are implemented by the AI core.





DVPP —— Brief Review

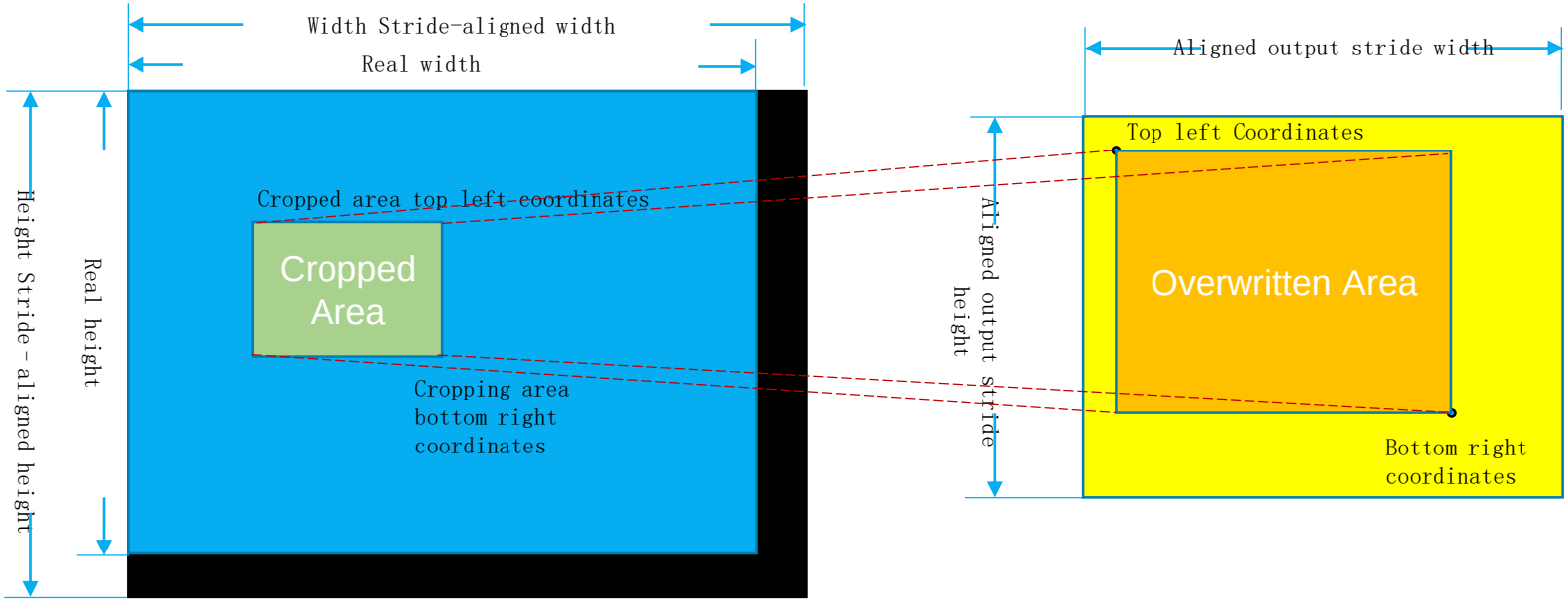
DVPP(Digital Video Pre-Processor) : image/video encoding, decoding, resizing, cropping, format conversion.



DVPP	Digital Vision Pre-Process
VPC	Vision Preprocess Core
JPEGD	JPEG Decode
JPEGE	JPEG Encoder
PNGD	Portable Network Graphics Decoder
VDEC	Video Decoder
VENC	Video Encoder

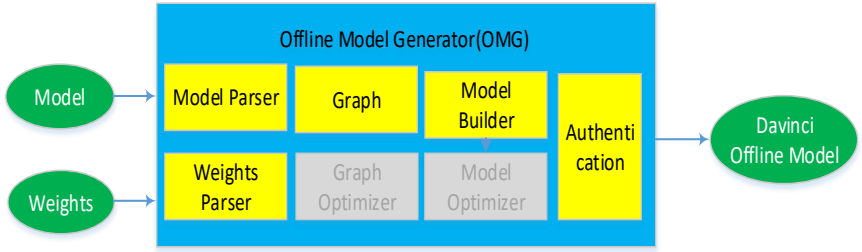
DVPP ——VPC Common Operation

VPC Common Operation: Crop+Resize:



Conversion Capabilities

Open-Source Framework	Input File	Storage Format
Caffe	*.prototxt, network structure file	Protobuf format
	*.caffemodel, weight file	Binary file
~~Caffe2~~	predict_net.pb, network definition file	Protobuf format
	All input data of the init_net.pb network, including all weight data and input data description of the first operator	Protobuf format
~~Mxnet~~	xx-symbol.json, network structure file	JSON format
	xx.params, weight file including the data, data type, and data format of the weight node	Binary file
Tensorsfolw	*.pb: The network model and weight data are in the same file.	Protobuf format



- The OMG adapts to different network files and weight files in various frameworks and converts them into offline files in DaVinci format for the OME to use.
- The OMG is independently executed offline on the host (Linux Ubuntu) and is provided to the IDE as an OMG executable program. The OMG can be called by command lines.

OMG —— CMD Introduction

Sample cmd:

./omg --model=./alexnet.prototxt --weight=./alexnet.caffemodel --framework=0 --output=./domi

./omg --model=$HOME/ResNet18_deploy.prototxt --weight= $HOME/ResNet18_model.caffemodel --framework=0 --output=$HOME/outdata/ResNet18 --input_shape= "data:1 ,3, 224, 224" --insert_op_conf= ./aipp_con.cfg

```
omg: usage: ./omg <args>
example:
./omg --model=./alexnet.prototxt --weight=./alexnet.caffemodel
--framework=0 --output=./domi
aguments explain:
  --model             Model file
  --weight            Weight file. Required when framework is Caffe
  --framework         Framework type(0:Caffe; 3:Tensorflow)
  --output            Output file path&name(needn't suffix, will add .om automatically)
  --encrypt_mode      Encrypt flag. 0: encrypt; -1(default): not encrypt
  --encrypt_key       Encrypt_key file
  --certificate       Certificate file
  --hardware_key      ISV file
  --private_key       Private key file
  --input_shape       Shape of input data. E.g.: "input_name1:n1,c1,h1,w1;input_name2:n2,c2,h2,w2"
  --h/help            Show this help message
  --cal_conf          Calibration config file
  --insert_op_conf    Config file to insert new op
  --op_name_map       Custom op name mapping file
  --plugin_path       Custom op plugin path. Default value is: "./plugin". E.g.: "path1;path2;path3".
                      Note: A semicolon(;) cannot be included in each path, otherwise the resolved path will not match the expected one.
  --om                The model file to be converted to json
  --json              The output json file path&name which is converted from a model
  --mode              Run mode. 0(default): model => davinci; 1: framework/davinci model => json; 3: only pre-check
  --target            Target platform. (mini)
  --out_nodes         Output nodes designated by users. E.g.: "node_name1:0;node_name1:1;node_name2:0"
  --input_format      Format of input data. E.g.: "NCHW"
  --check_report      The pre-checking report file. Default value is: "check_result.json"
  --input_fp16_nodes  Input node datatype is fp16 and format is NCHW. E.g.: "node_name1;node_name2"
  --is_output_fp16    Net output node datatype is fp16 and format is NCHW, or not. E.g.: "false,true,false,true"
  --ddk_version       The ddk version. E.g.: "x.y.z.Patch.B350"
  --net_format        Set net prior format. ND: select op's ND format preferentially; 5D: select op's 5D format preferentially
  --output_type       Set net output type. Support FP32 and UINT8
  --fp16_high_prec    FP16 high precision. 0(default): not use fp16 high precision; 1: use fp16 high precision
```

Common Paras:

--model	path to original model file
--weight	path to weight file (Caffe)
--framework	0: caffe 3: tensorflow
--output	output om file path
--plugin_path	customized Operator path
--insert_op_conf	aipp config file path
--input_shape	input data shape。
--fp16_high_prec	generate FP16 model
--mode	0: generate davinci model 1: om model to json 3: only pre-check

For other parameters, please refer to Atlas200DK model conversion guide

MindSpore: All-scenario AI computing framework

AI applications

MindSpore

Unified APIs for all scenarios

Automatic differentiation	Automatic parallelization	Automatic tuning

MindSpore intermediate representation (IR) for computation graphs

On-device execution	Pipeline parallelism	Deep graph optimization

Device-edge-cloud cooperative distributed architecture (for deployment, scheduling, communications, etc.)

Processors: Ascend, GPU, CPU

MindSpore

Easy development: AI Algorithm As Code

Efficient execution: Optimized for Ascend, GPU support

Flexible deployment: On-demand cooperation across all scenarios

ModelArts: Full-pipeline model production services

ModelArts

Supports full pipeline – From data collection and model development to model training and deployment

4,000+ training tasks per day (total of 32,000 training hours)

Visual tasks: **85%**; audio tasks: **10%**; ML tasks: **5%**

30,000+ developers

Software + hardware co-optimization: Stronger performance

AI computing challenges

Complex computing

- Scalar, vector, and tensor computing
- Hybrid precision computing
- Parallelism between data augmentation and minibatch computing
- Parallelism between gradient aggregation and minibatch computing

Diverse computing power

- CPUs, GPUs, and Ascend processors
- Diverse computing units: scalar, vector, and tensor

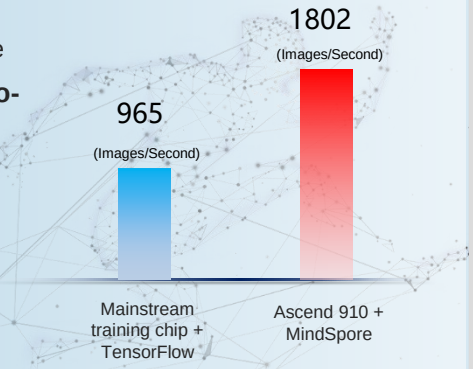
MindSpore

Framework optimization

- Pipeline parallelism
- Cross-layer memory reuse

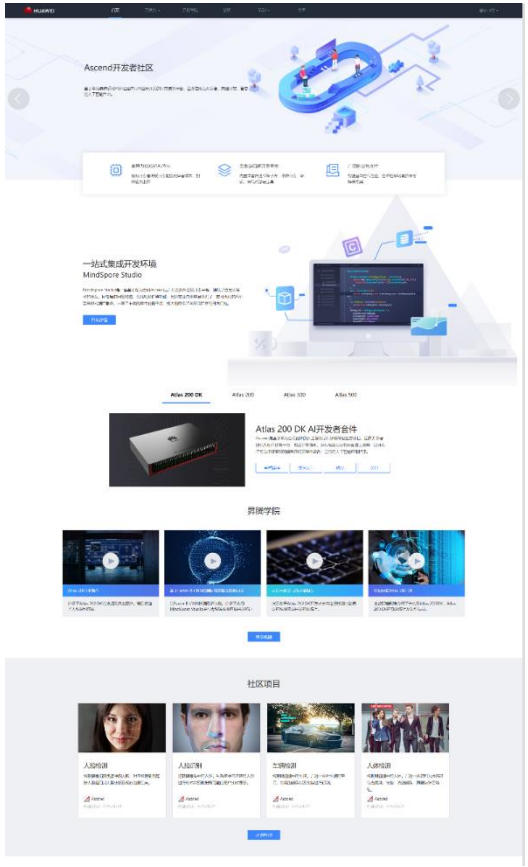
Software + hardware co-optimization

- On-device execution
- Deep graph optimization



- ResNet 50 V1.5
- ImageNet 2012
- Based on optimal batch sizes

Ascend Developer Community: End-to-End Services for Developers



Ascend developer portal

https://ascend.huawei.com

Unified channel for open capability release and operation

One-stop support services

Technical documents, online resources, development tools, video tutorials, support services, interactive communities, and information release

Developer-centric experience

Provides rich, friendly, easy-to-use, and quick portal experience for developers

Ascend Eco-Systems in China

Huawei AI Developer Support Program

Developer Challenge

Developer Technical Salon

- Public cloud voucher
- Certification course ticket
- Atlas developer kit
- HUAWEI CLOUD MVP
- Prize money

Developed 50,000 developers and 100+ startups in 2019

AI Talent Development Program

The first batch of courses co-created with colleges

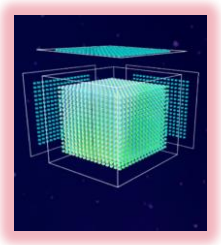
Shanghai Jiao Tong University
Zhejiang University
Tsinghua University
Peking University
Tianjin University
Renmin University of China
Beijing University of Posts and Telecommunications
Huazhong University of Science and Technology
Southern University of Science and Technology
Xidian University

Co-created courses & labs with 30+ colleges in 2019

We also already started in Canada:

- SFU： 2020 Spring Term Computer Vision class for Professional Master program;
- MakeUofT Hackathon: Feb 15 -16

Thank you.



Da Vinci

Ascend

Atlas

Bring digital to every person, home and organization for a fully connected, intelligent world.

HUAWEI