Machine Learning @ Microsoft

Stanford Scaled Machine Learning Conference

August 2nd 2016

Qi Lu, Application & Services Group, Microsoft

Agenda

- What We Do
  - History
  - Going forward
- How We Scale
  - CNTK
  - FPGA
  - Open Mind
- Q&A

What We Do

ML @ Microsoft: History

Answering questions with experience

1991	1997	2008	2009	2010	2014	2015
Microsoft Research formed	Hotmail launches	Bing maps launches	Bing search launches	Kinect launches	Skype Translator launches	Azure Machine Learning GA Office 365 Substrate HoloLens
	Which email is junk?	What's the best way home?	Which URLs are most relevant?	What does that motion "mean"?	What is that person saying?	What will happen next?

Machine learning is pervasive throughout Microsoft products

ML @ Microsoft: Going Forward

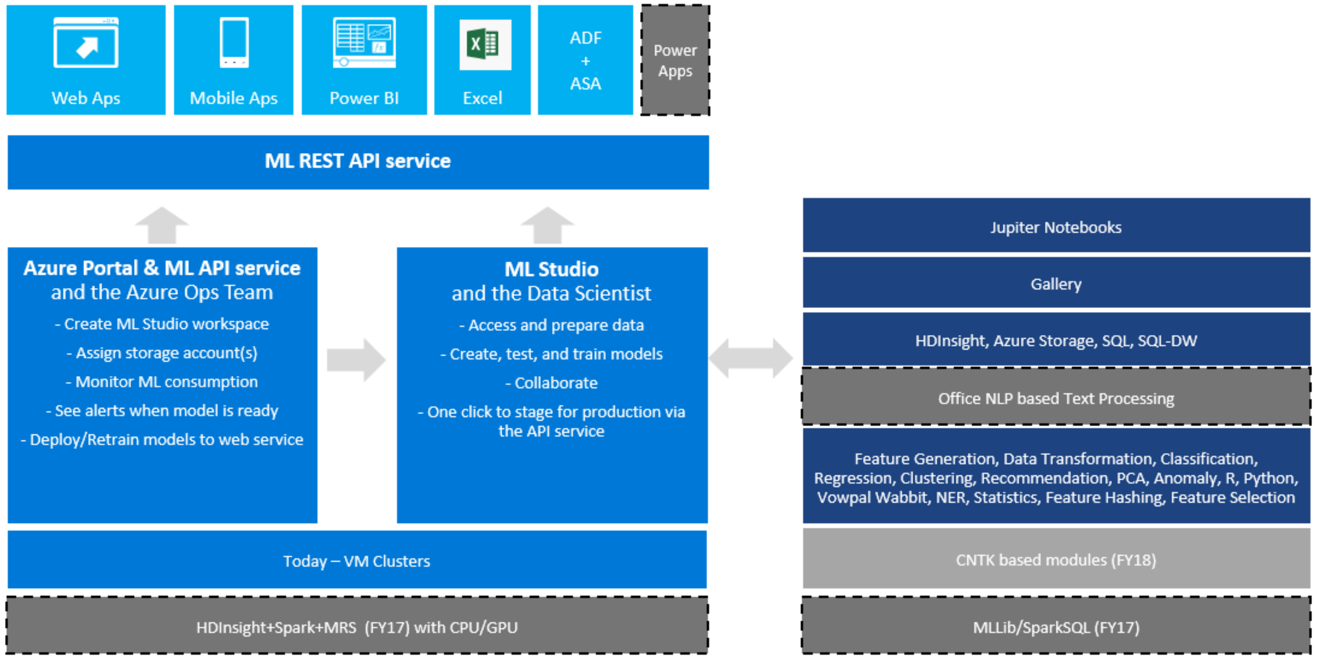
- Data => Model => Intelligence => Fuels of Innovation
- Applications & Services
  - Office 365, Dynamic 365 (Biz SaaS), Skype, Bing, Cortana
  - Digital Work & Digital Life
  - Models for: World, Organizations, Users, Languages, Context, …
- Computing Devices
  - PC, Tablet, Phone, Wearable, Xbox, Hololens (AR/VR), ….
  - Models for: Natural User Interactions, Reality, …
- Cloud
  - Azure Infrastructure and Platform
  - Azure ML Tools & Services
  - Intelligence Services

Machine Learning Building Blocks

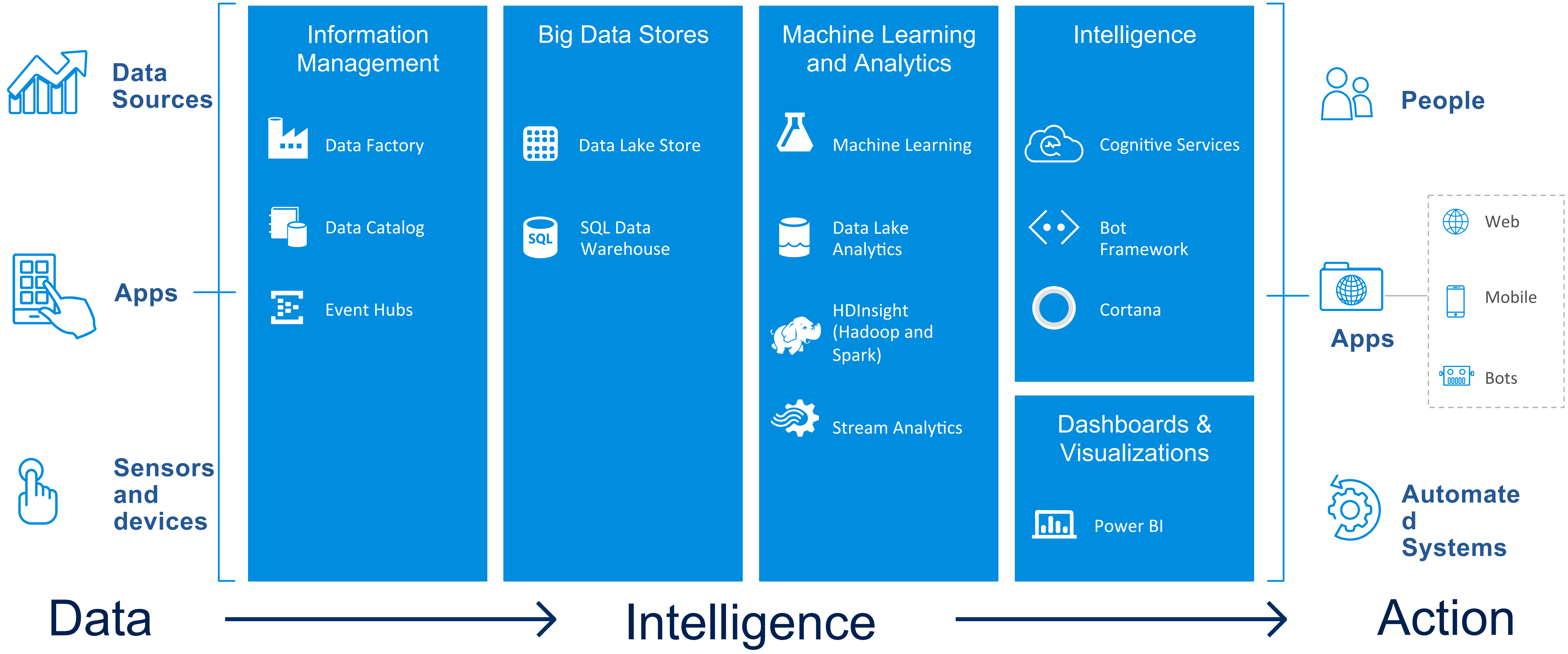
Azure ML (Cloud)	Microsoft R Server (On-Prem & Cloud)	Computational Network Toolkit	Cognitive APIs (Cloud Services)	HDInsight/Spark
Ease of use through Visual Workflows	Enterprise Scale & Performance	Designed for peak performance	See, hear, interpret, and interact	Open source Hadoop with Spark
Single click operationalization	Write Once, Deploy Anywhere	Works on CPU and GPU (single/multi)	Prebuilt APIs with CNTK and experts	Use Spark ML or MLLib using Java, Python, Scala or R
Expand reach with Gallery and marketplace	R Tools for Visual Studio IDE	Supports popular network types (FNN, CNN, LSTM, RNN)	Vision, Speech, Language, Knowledge,	Support for Zeppelin and Jupyter notebook
Integration with Jupyter Notebook	Secure/Scalable Operationalization	Highly Flexible – description language	Build and connect intelligent bots	Includes MRS over Hadoop or over Spark
Integration with R/ Python	Works with open source R	Used to build cognitive APIs	Interact with your users on SMS, text, email, Slack, Skype	Train on TBs of data
				Run large massively parallel compute and data jobs

Azure Machine Learning Services

- Ease of use tools with **drag/drop paradigm**, single click **operationalization**
- Built-in support for **statistical functions**, data **ingest**, **transform, feature generate/select, train, score, evaluate** for tabular data and text across **classification, clustering, recommendation, anomaly**
- Seamless **R/Python** integration along with support for **SQL lite** to filter, transform
- **Jupyter** Notebooks for data exploration and **Gallery** extensions for quick starts
- Modules for **text preprocessing**, key phrase extraction, language detection, n-gram generation, LDA, compressed feature hash, stats based anomaly
- **Spark/HDInsight/MRS** Integration
- **GPU** support
- New geographies
- Compute **reservation**



Intelligence Suite



Cognitive Services

Vision	Speech	Language	Knowledge	Search
Computer vision	Speaker recognition	Text analytics	Academic knowledge	Bing search API
Face	Speech	Bing spell check	Entity linking service	Bing image search API
Emotion	Custom recognition	Web language model	Knowledge exploration service	Bing video search API
Video		Linguistic analysis	Recommendations	Bing news search API
		Language understanding		Bing auto suggest API
		Translator		

How We Scale

Key Dimensions of Scaling

- Data volume / dimension
- Model / algorithm complexity
- Training / evaluation time
- Deployment / update velocity
- Developer productivity / innovation agility
- Infrastructure / platform
- Software framework / tool
- Data set / algorithm

How We Scale Example: CNTK

CNTK: Computational Network Toolkit

- CNTK is Microsoft's open-source, cross-platform toolkit for learning and evaluating models especially deep neural networks
- CNTK expresses (nearly) arbitrary neural networks by composing simple building blocks into complex computational networks, supporting common network types and applications
- CNTK is production-deployed: accuracy, efficiency, and scales to multi-GPU/multi-server

CNTK Development

- Open-source development model inside and outside the company
  - Created by Microsoft Speech researchers 4 years ago; open-sourced in early 2015
  - On GitHub since Jan 2016 under permissive license
  - Nearly all development is out in the open
- Driving applications: Speech, Bing, Hololens, MSR research
  - Each team have full-time employees actively contribute to CNTK
  - CNTK trained models are tested and deployed in production environment
- External contributions
  - e.g., from MIT and Stanford
- Platforms and runtimes
  - Linux, Windows, .Net, docker, cudnn5
  - Python, C++, and C# APIs coming soon

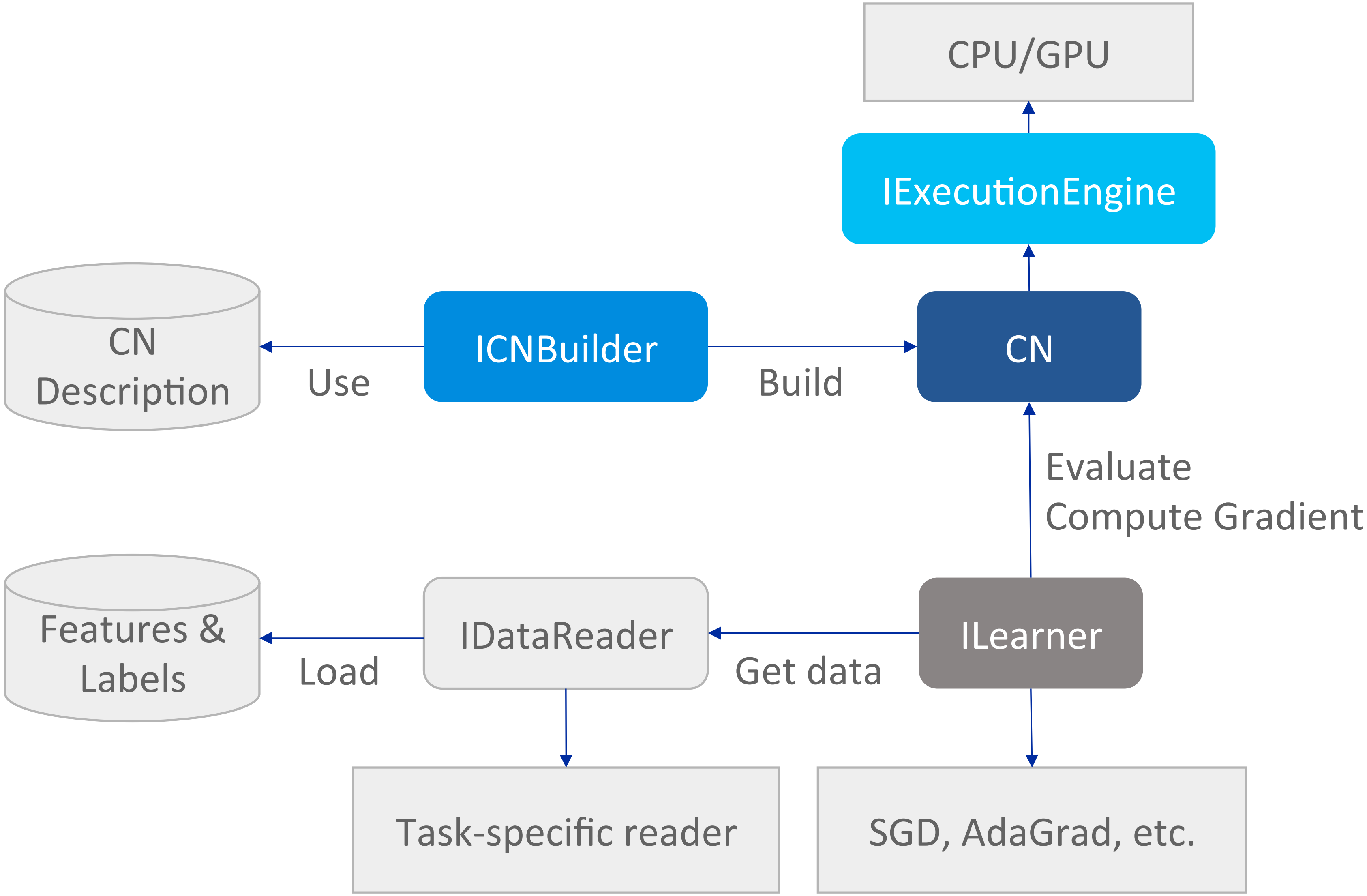
CNTL Design Goals & Approach

- A deep learning framework that balances
  - Efficiency: can train production systems as fast as possible
  - Performance: can achieve best-in-class performance on benchmark tasks for production systems
  - Flexibility: can support a growing and wide variety of tasks such as speech, vision, and text; can try out new ideas very quickly
- Lego-like composability
  - Support a wide range of networks
  - E.g. Feed-forward DNN, RNN, CNN, LSTM, DSSM, sequence-to-sequence
- Evolve and adapt
  - Design for emerging prevailing patterns

Key Functionalities & Capabilities

- Supports
  - CPU and GPU with a focus on GPU Cluster
  - Automatic numerical differentiation
  - Efficient static and recurrent network training through batching
  - Data parallelization within and across machines, e.g., 1-bit quantized SGD
  - Memory sharing during execution planning
- Modularization with separation of
  - Computational networks
  - Execution engine
  - Learning algorithms
  - Model description
  - Data readers
- Model descriptions via
  - Network definition language (NDL) and model editing language (MEL)
  - Brain Script (beta) with Easy-to-Understand Syntax

Architecture

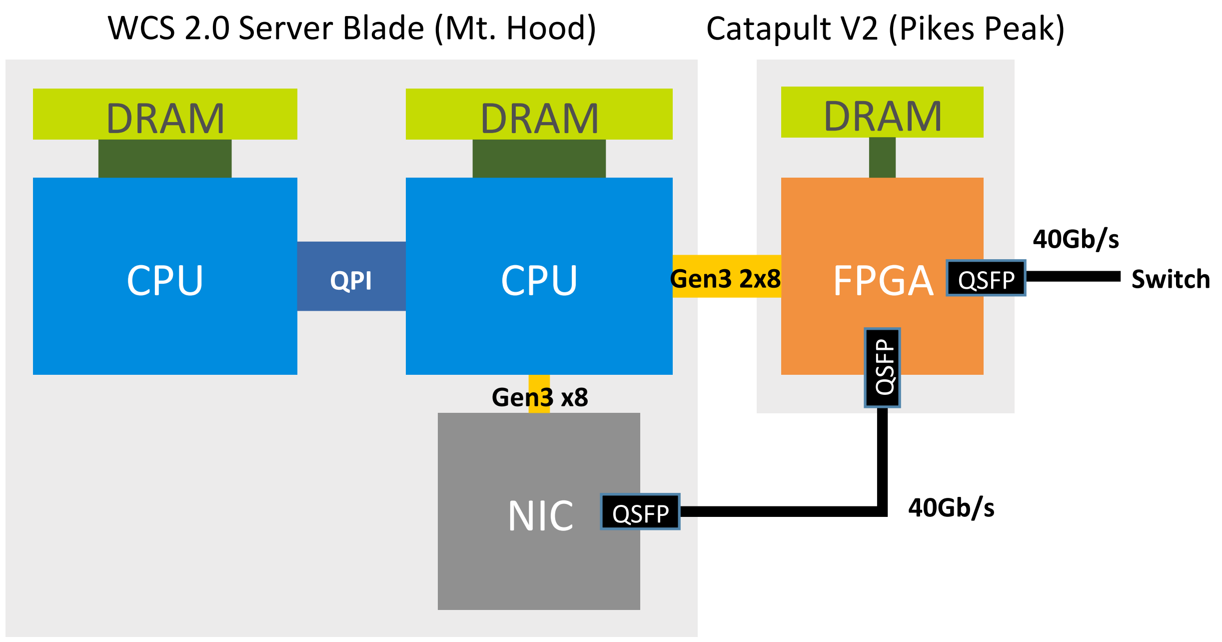


Roadmap

- CNTK as a library
  - More language support: Python/C++/C#/.Net
- More expressiveness
  - Nested loops, sparse support
- Finer control of learner
  - SGD with non-standard loops, e.g., RL
- Larger model
  - Model parallelism, memory swapping, 16-bit floats
- More powerful CNTK service on Azure
  - GPUs soon; longer term with cluster, container, new HW (e.g., FPGA)

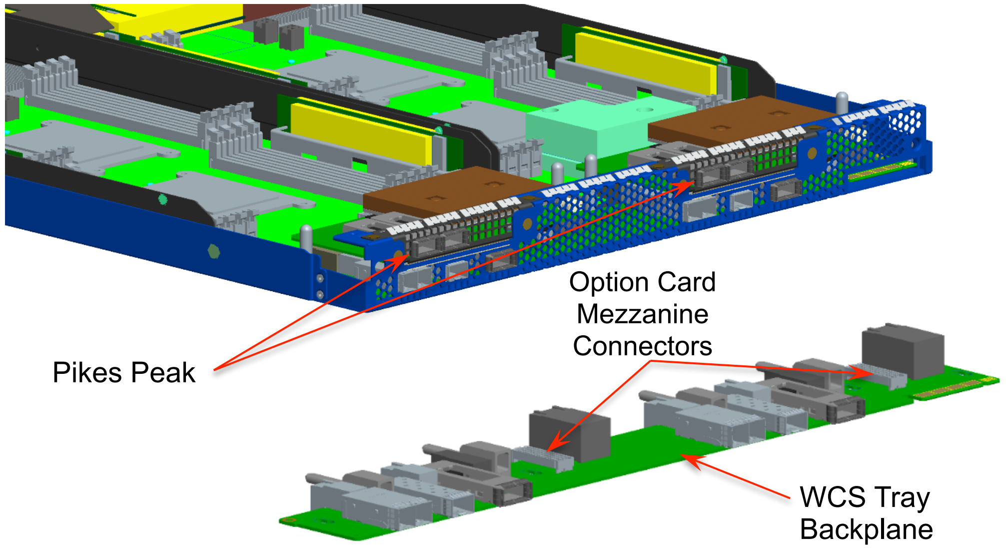
How We Scale Example: FPGA

Catapult v2 Architecture



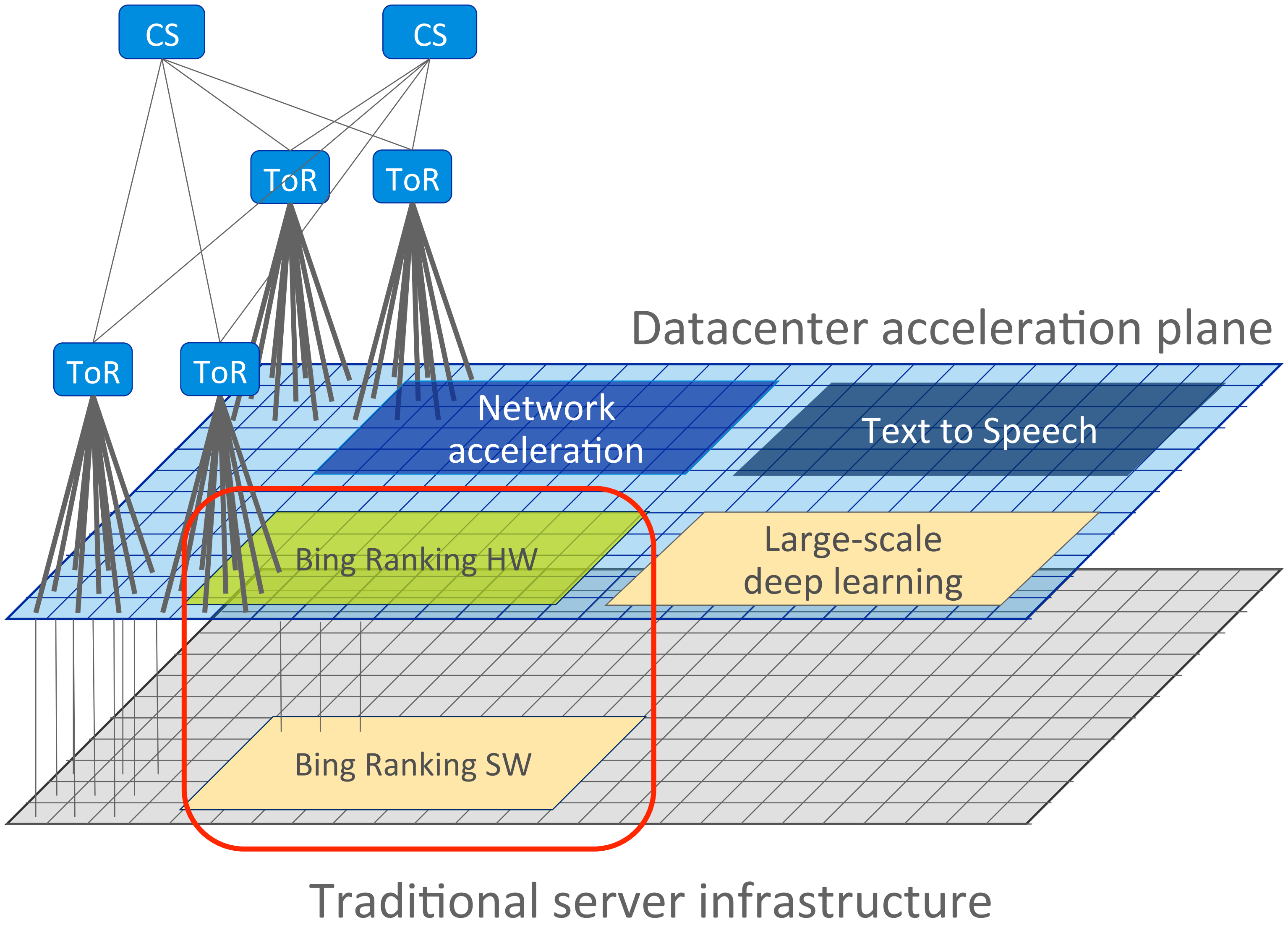
Catapult WCS Mezz card
(Pike's Peak)

WCS Gen4.1 Blade with Mellanox NIC and Catapult FPGA



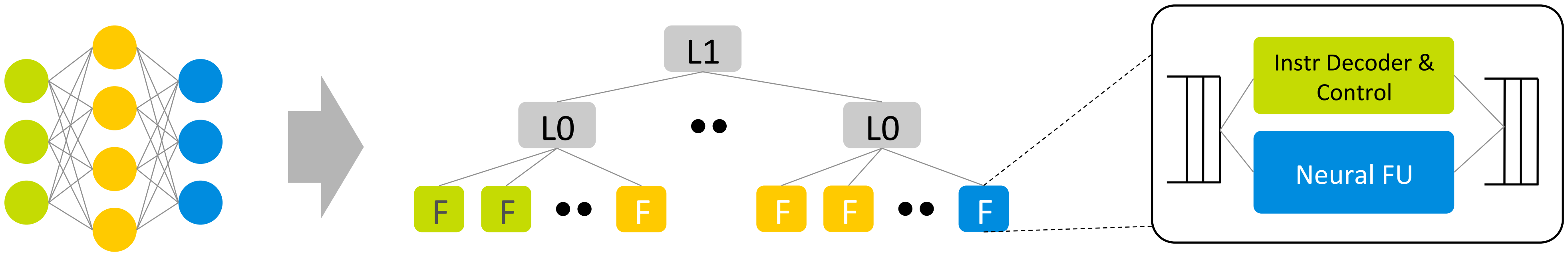
- Gives substantial acceleration flexibility
  - Can act as a local compute accelerator
  - Can act as a network/storage accelerator
  - Can act as a remote compute accelerator

Configurable Clouds



- Cloud becomes network + FPGAs attached to servers
- Can continuously upgrade/change datacenter HW protocols (network, storage, security)
- Can also use as an application acceleration plane (Hardware Acceleration as a Service (HaaS)
- Services communicate with no SW intervention (LTL)
- Single workloads (including deep learning) can grab 10s, 100s, or 1000s of FPGAs
- Can create service pools as well for high throughput

Scalable Deep Learning on FPGAs



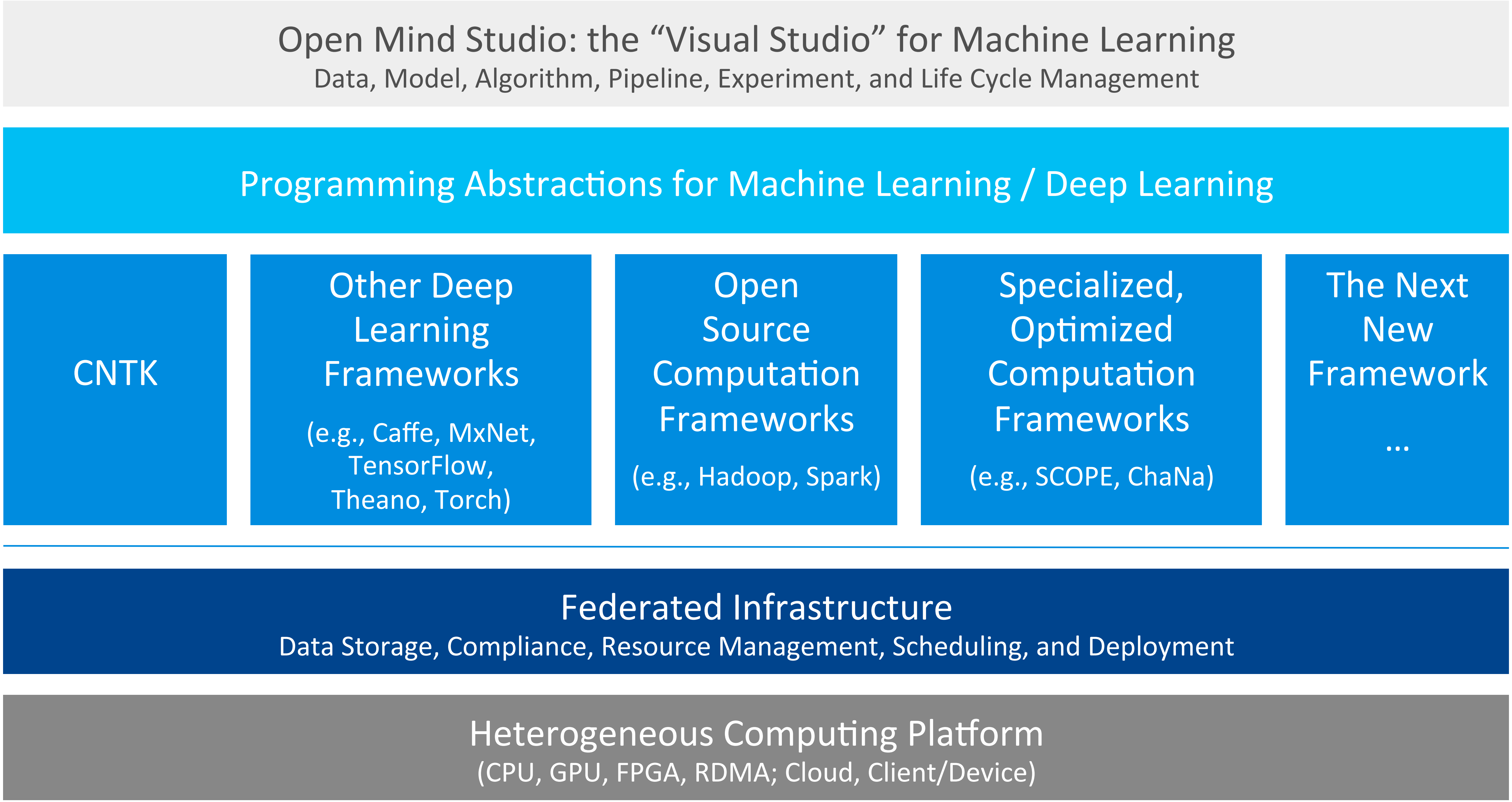
NN Model

FPGAs over HaaS

Scale ML Engine

- Scale ML Engine: a flexible DNN accelerator on FPGA
  - Fully programmable via software and customizable ISA
  - Over 10X improvement in energy efficiency, cost, and latency versus CPU
- Deployable as large-scale DNN service pools via HaaS
  - Low latency communication in few microseconds / hop
  - Large scale models at ultra low latencies

How We Scale Example: Open Mind


Open Mind Studio: the “Visual Studio” for Machine Learning
Data, Model, Algorithm, Pipeline, Experiment, and Life Cycle Management
Programming Abstractions for Machine Learning / Deep Learning
CNTK
Other Deep Learning Frameworks
(e.g., Caffe, MxNet, TensorFlow, Theano, Torch)
Open Source Computation Frameworks
(e.g., Hadoop, Spark)
Specialized, Optimized Computation Frameworks
(e.g., SCOPE, ChaNa)
The Next New Framework
…
Federated Infrastructure
Data Storage, Compliance, Resource Management, Scheduling, and Deployment
Heterogeneous Computing Platform
(CPU, GPU, FPGA, RDMA; Cloud, Client/Device)

ChaNa:RDMA-Optimized Computation Framework

- Focus on faster network
  - Compact memory representation
  - Balanced parallelism
  - Highly optimized RDMA-aware communication primitives
  - Overlapping communication and computation
- An order of magnitude improvement in early results
  - Over existing computation frameworks (with TCP)
  - Against several large-scale workloads in production

Programming Abstraction for Machine Learning

- Graph Engines for Distributed Machine Learning
  - Automatic system-level optimizations
  - Parallelization and distribution
  - Layout for efficient data access
  - Partitioning for balanced parallelism
- Promising early results
  - Simplification of distributed ML programs via high level abstractions
  - About 70-80% reduction in code
    - Relative to ML systems such as Petuum, Parameter Server
    - Matrix Factorization for recommendation system
    - Latent Dirichlet Allocation for topic modeling

Q&A

Thank You!