Enabling Cross-platform Deep Learning Applications with Intel OpenVINO™

Yury Gorbachev

May 22, 2018

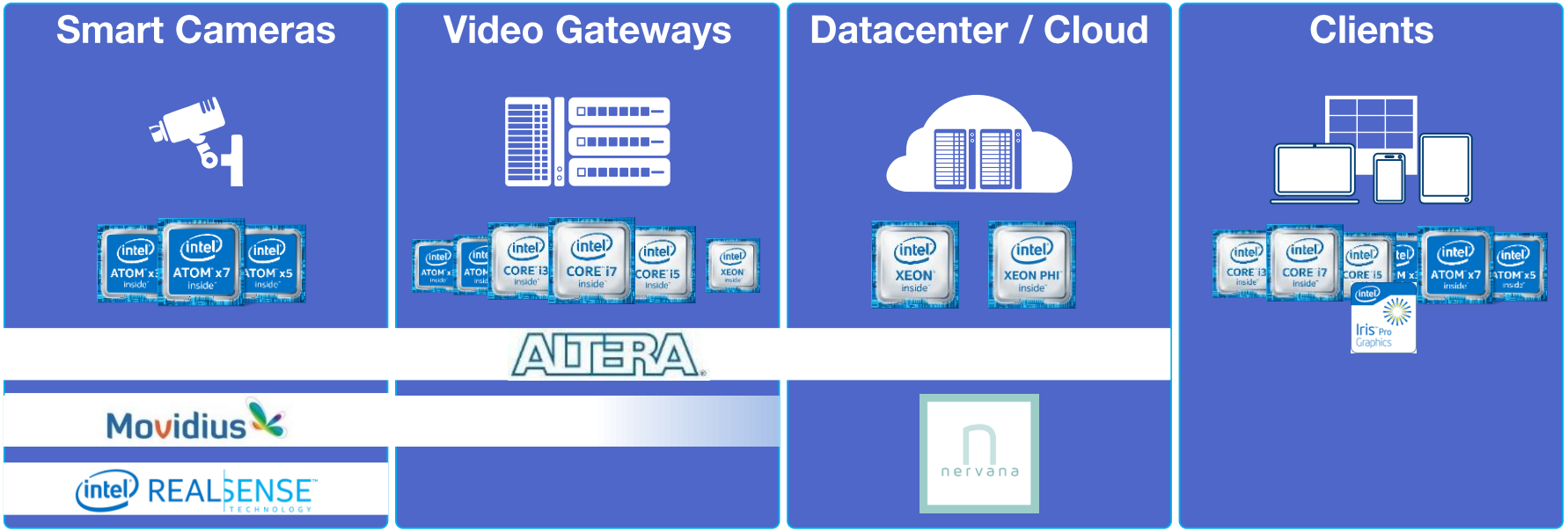
Customer demands and industry trends

- Faster TTM: lower porting burden from algo prototype to deployment
- Cross-platform portability/future-proofing of applications: Deploy the same algorithm on platforms with different DL accelerators/different network nodes (e.g. smart cameras, NVRs, cloud)
- Smarten up already installed products
  - Add functionality incrementally (increase complexity)
  - Update existing algorithms (improve quality)
- Scale processing with increasing demand or complexity
- Differentiate with algorithms and intelligence

Hurdles

- Deployment in different environments causes portability issues
  - HW/SW architectures differ, thus application redesign may be required
  - Rapidly moving target as number of platforms grows
  - DL model retraining/redesign may be required
- CV/DL domain is complex and features a high entry threshold
  - Low performing academic results are frequently used in production
- Existing Deep Learning solutions are mostly training oriented

Intel HW portfolio



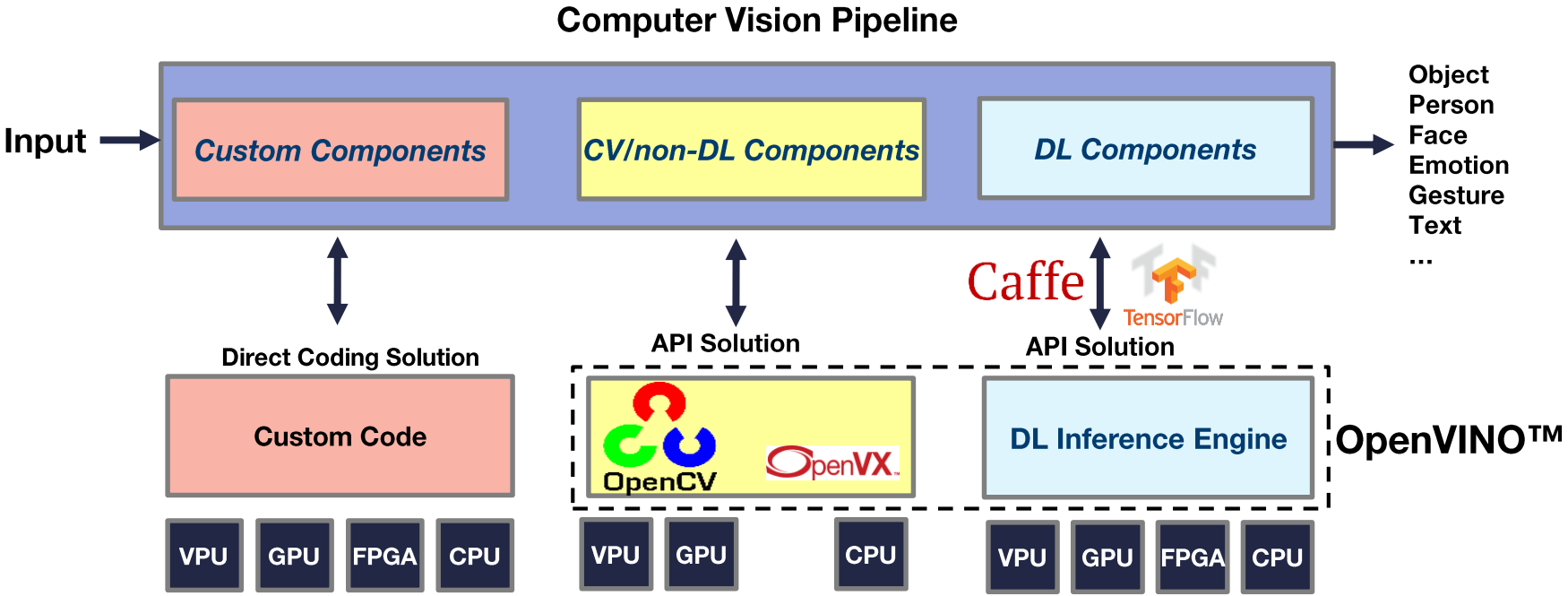
Intel Delivers

- End-to-End Architecture
- X86 software portability & ecosystem
- Suite of SDKs and tools

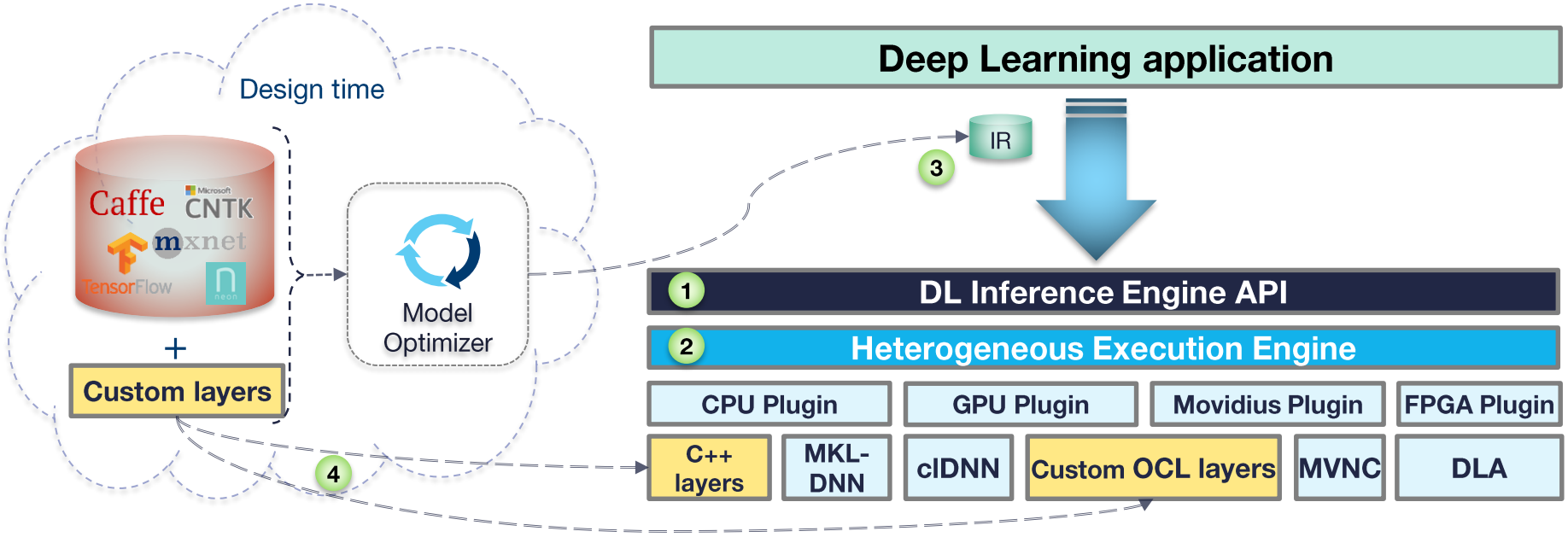
OpenVINO™ Toolkit for best CV/DL applications

- Development toolkit for high performance CV and DL inference
  - Solution for application designers
  - No training overhead or specifics, minimal footprint, highly portable code
- Set of libraries to solve CV/DL deployment problems
  - Fastest OpenCV build
  - Certified OpenVX implementation
  - Deep Learning Inference Engine
- Provides access to all accelerators and heterogeneous execution model
  - Intel CPU, CPU w/integrated graphics
  - Vision Processing Unit (VPU) and FPGA

Computer Vision using Intel OpenVINO™ Toolkit



Deep Learning Inference Engine (IE)



1. Single API solution across accelerators
2. Heterogeneous network execution across accelerators
3. Framework independent lightweight internal representation
4. Customizations in C++ and OpenCL languages

Cross platform portability

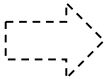
- Original training framework is not required for execution
- Same API across all Intel accelerator offerings
  - No need for application redesign when deploying on different target
- Consistent DL models accuracy and functionality across targets
  - No model retraining is required
- Comprehensive validation suite

Additional portability benefits

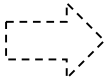
- Design future-proof products that use evolving Intel DL accelerators, including the ones under development

- Convenient and available targets for design and validation
  - Very easy to troubleshoot on traditional desktops

Create application using Inference Engine API



Design and validate on CPU/GPU clusters



Deploy on Movidius/FPGA Targets

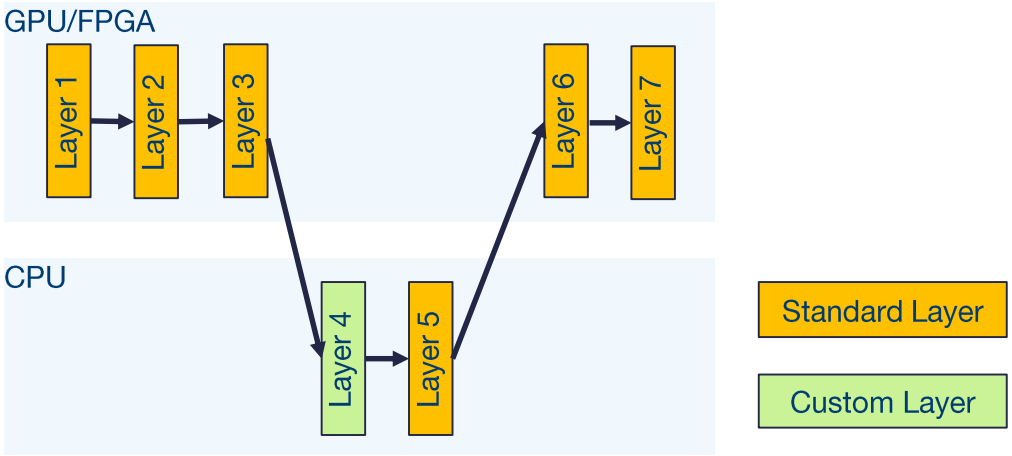
Customization and source availability

- Custom layers extension mechanism supported via API
  - New topologies and features are easy to support
  - Keep your own intellectual property protected

- MKL-DNN and clDNN back-ends available in source form
- Most of the layers are distributed in source form
  - Easy to modify for your needs, remove unnecessary features
  - Use as a sample for faster time to market
  - C++ for CPU, OpenCL for other IPs

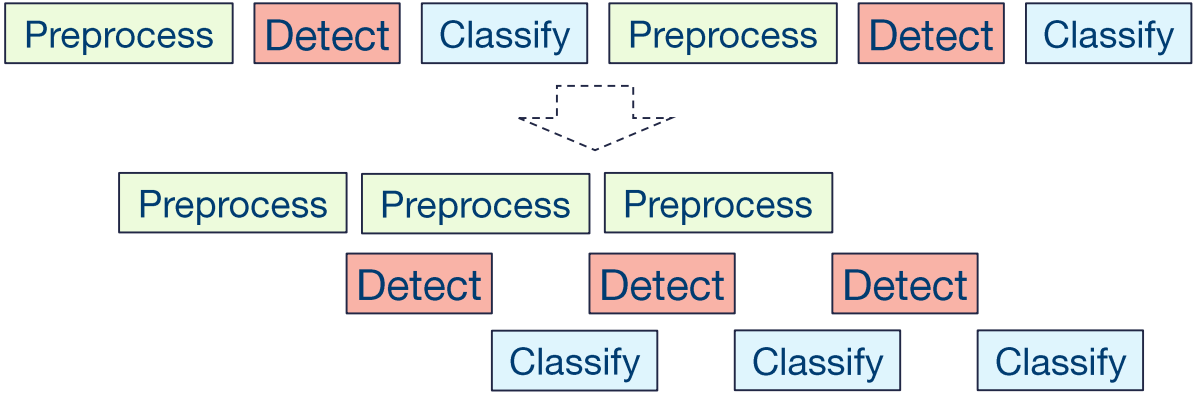
Heterogeneous execution

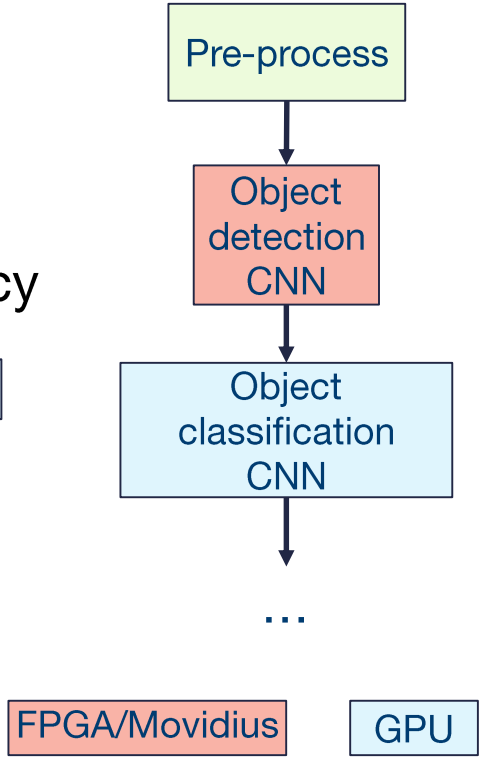
- Split graph execution between multiple targets
  - Custom layers via CPU code if needed
  - Better performance for certain functions (e.g. GPU vs CPU)



Asynchronous execution

- Asynchronous execution
  - Hide transfer latency (e.g. FPGA offload)
  - Perform parallel compute on multiple targets
  - Efficient SoC utilization and lower pipeline latency



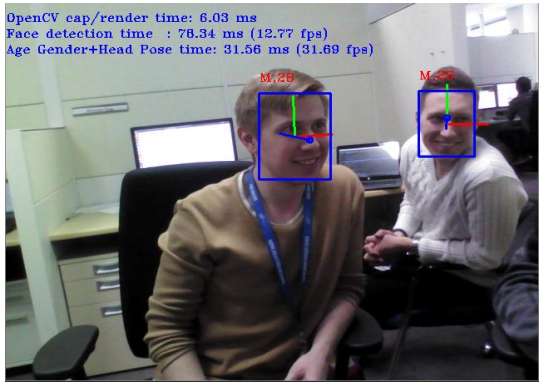


High quality DL models (IoT Model Zoo)

- Free reference models for Deep Learning Inference Engine
  - Object Detection (Face, People, Vehicles, etc.)
  - Object Analysis (Facial attributes, Head Pose, Vehicle attributes)
- Superior performance on Intel
  - Core™ i5 CPU: SSD 300 (6 fps) vs. People Detection Model (60 fps)
- Significant reduction in development efforts, no dataset & training needed
- Squeeze-in more functions thanks to efficient models!

Samples

- Basic samples to facilitate API understanding
  - Classification, object detection, segmentation
  - Target selection via command line
- Extended samples using Model Zoo
  - Face analysis, Security camera sample
- Interworking between Media SDK, OpenCV, DL IE
- Automated public models downloader script



Resources and useful links

- OpenVINO™: https://software.intel.com/en-us/openvino-toolkit
- Support forum: https://software.intel.com/en-us/forums/computer-vision
- Visit our demo booth, ask questions

- E-mail me: yury.gorbachev@intel.com

Thank you