Sujata Tibrewala

OneAPI Community Development Manager
Intel

SUJATA.TIBREWALA@INTEL.COM

@SUJATATIBRE

Hello
Alessandro de Oliveira Faria

Founder of the Company
OITI TECHNOLOGIES

Making banking secure via bio metrics application built using oneAPI and DPC++ based on SYCL/C++

Notices and Disclaimers

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Intel technologies may require enabled hardware, software or service activation.

No product or component can be absolutely secure.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

- Searches and compares in more than 20 million in less than 4 seconds

- 100% in cloud

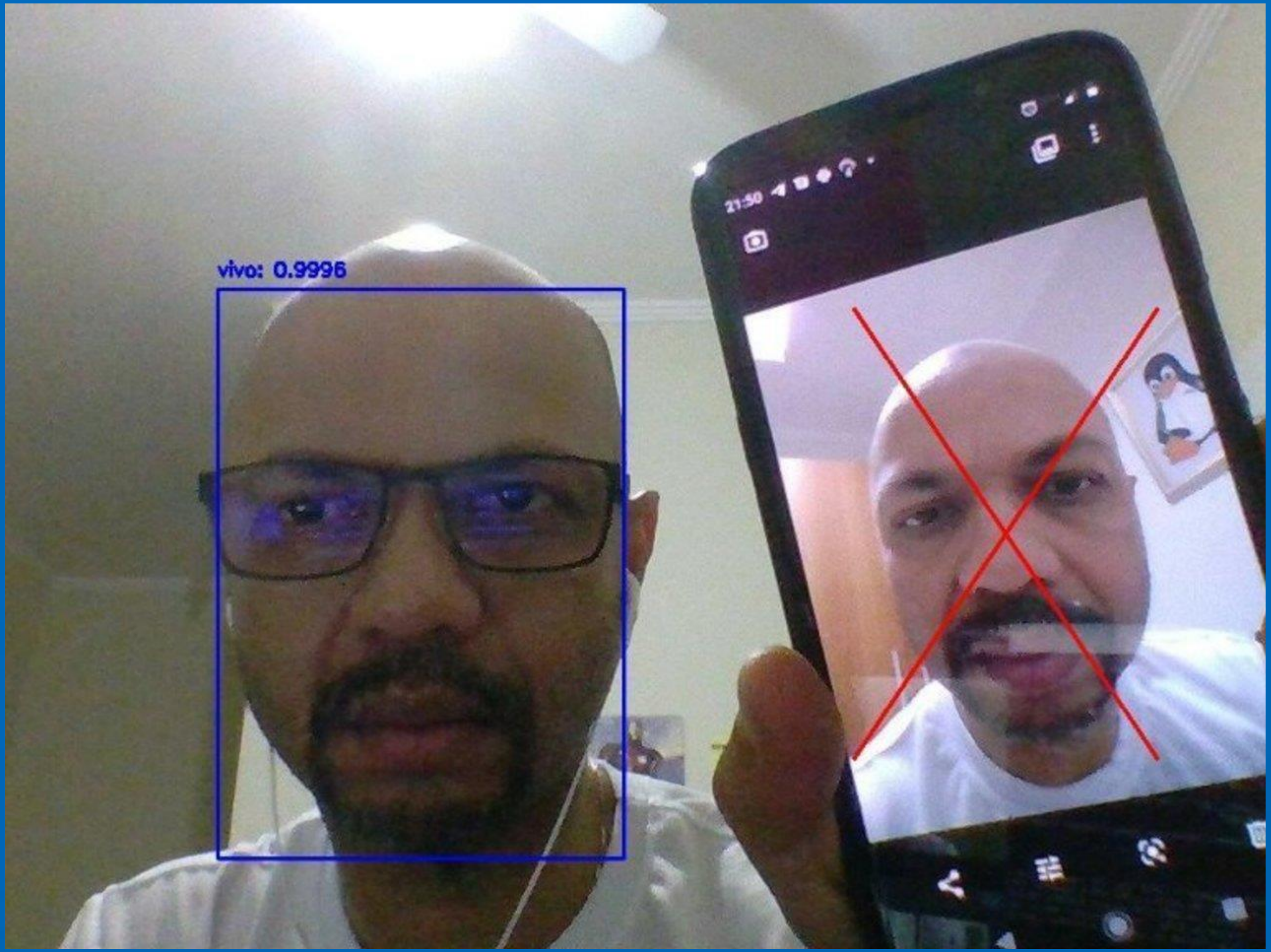
- 0,0000000007 False Accept Rate

- Fighst Fraud and Authenticate Users

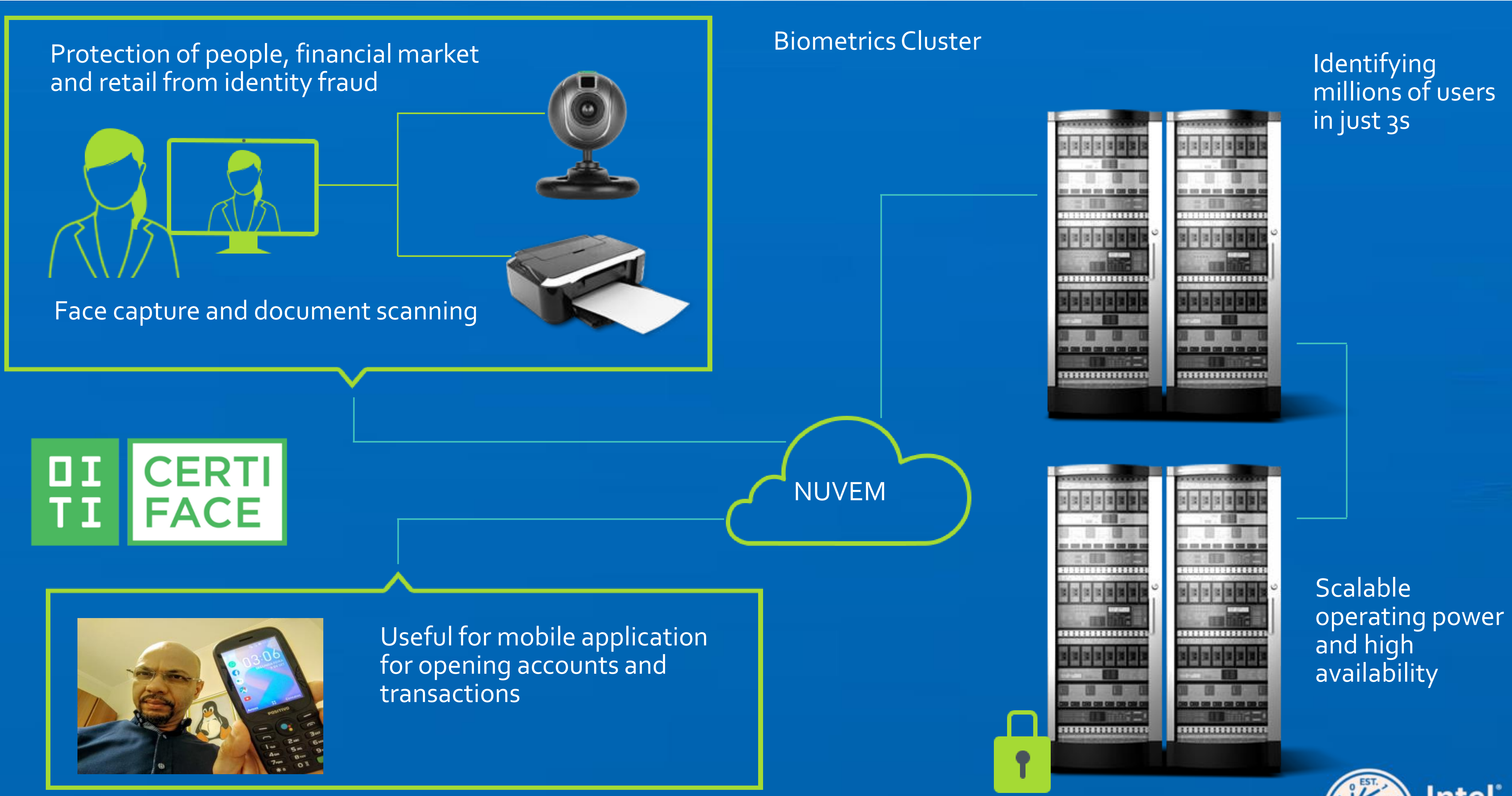
- Robust Algorithm (low image quality)

OITI CERTIFACE

Intel® Software Innovator



Intel® Software Innovator



Biometrics Cluster

Protection of people, financial market and retail from identity fraud

Face capture and document scanning

OITI CERTIFACE

NUVEM

Identifying millions of users in just 3s

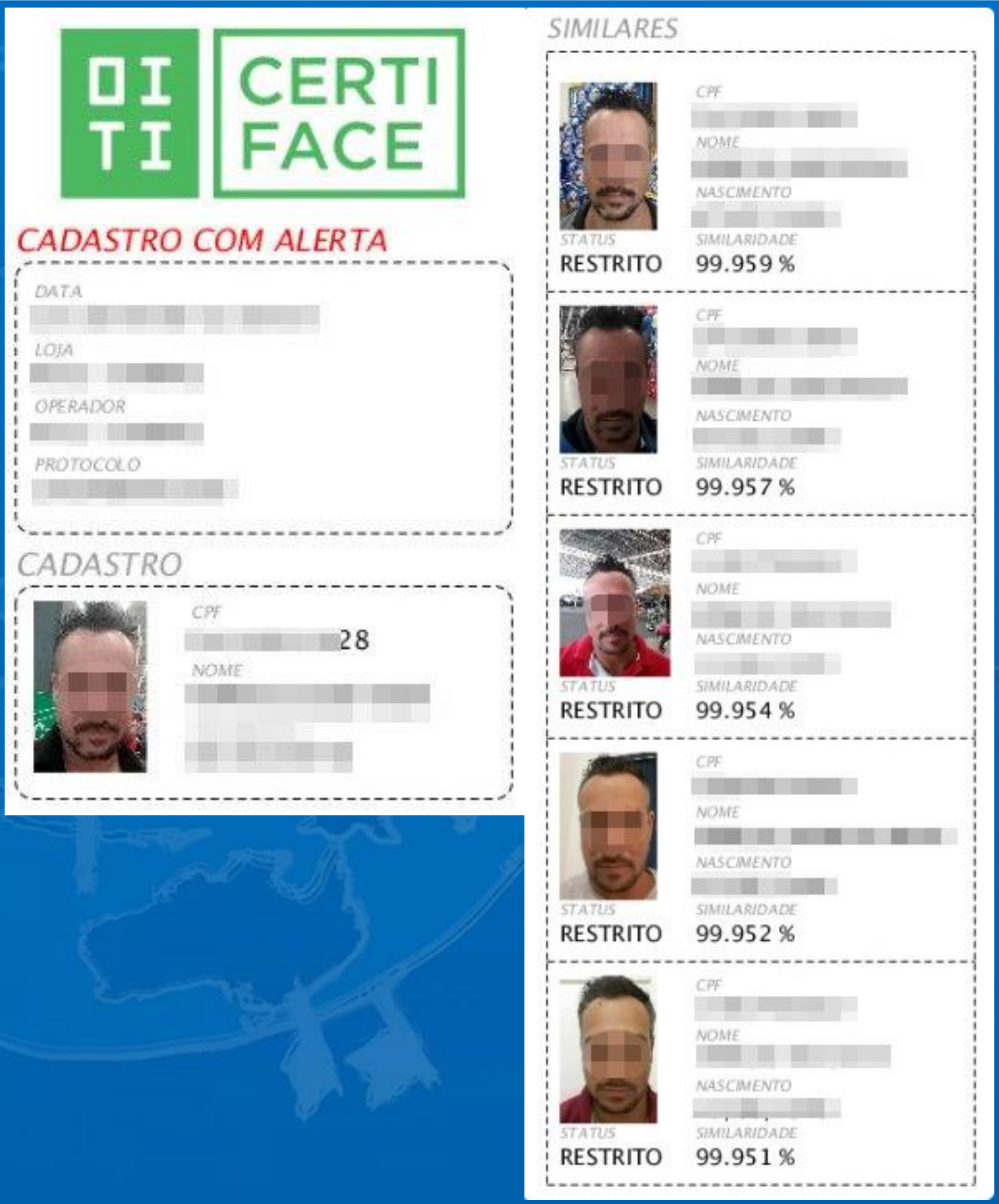
Scalable operating power and high availability

Useful for mobile application for opening accounts and transactions

Intel Software Innovator

Certiface protecst honest people, financial market and retail from fraud

When Certiface finds a linked face with many different documents (Tax ID), in the centralized base with millions of users, the system triggers an alert in real time



CADASTRO COM ALERTA

DATA [redacted]

LOJA [redacted]

OPERADOR [redacted]

PROTOCOLO [redacted]

CADASTRO

CPF [redacted]28

NOME [redacted]

SIMILARES

CPF [redacted]
NOME [redacted]
NASCIMENTO [redacted]
STATUS RESTRITO
SIMILARIDADE 99.959 %

CPF [redacted]
NOME [redacted]
NASCIMENTO [redacted]
STATUS RESTRITO
SIMILARIDADE 99.957 %

CPF [redacted]
NOME [redacted]
NASCIMENTO [redacted]
STATUS RESTRITO
SIMILARIDADE 99.954 %

CPF [redacted]
NOME [redacted]
NASCIMENTO [redacted]
STATUS RESTRITO
SIMILARIDADE 99.952 %

CPF [redacted]
NOME [redacted]
NASCIMENTO [redacted]
STATUS RESTRITO
SIMILARIDADE 99.951 %

Intel® Software Innovator

Robust Algorithm

Certiface uses several algorithms (proprietary and opensource) to process a robust solution based on computer vision.

Even with dark, low-quality photos, certiface can process facial biometrics

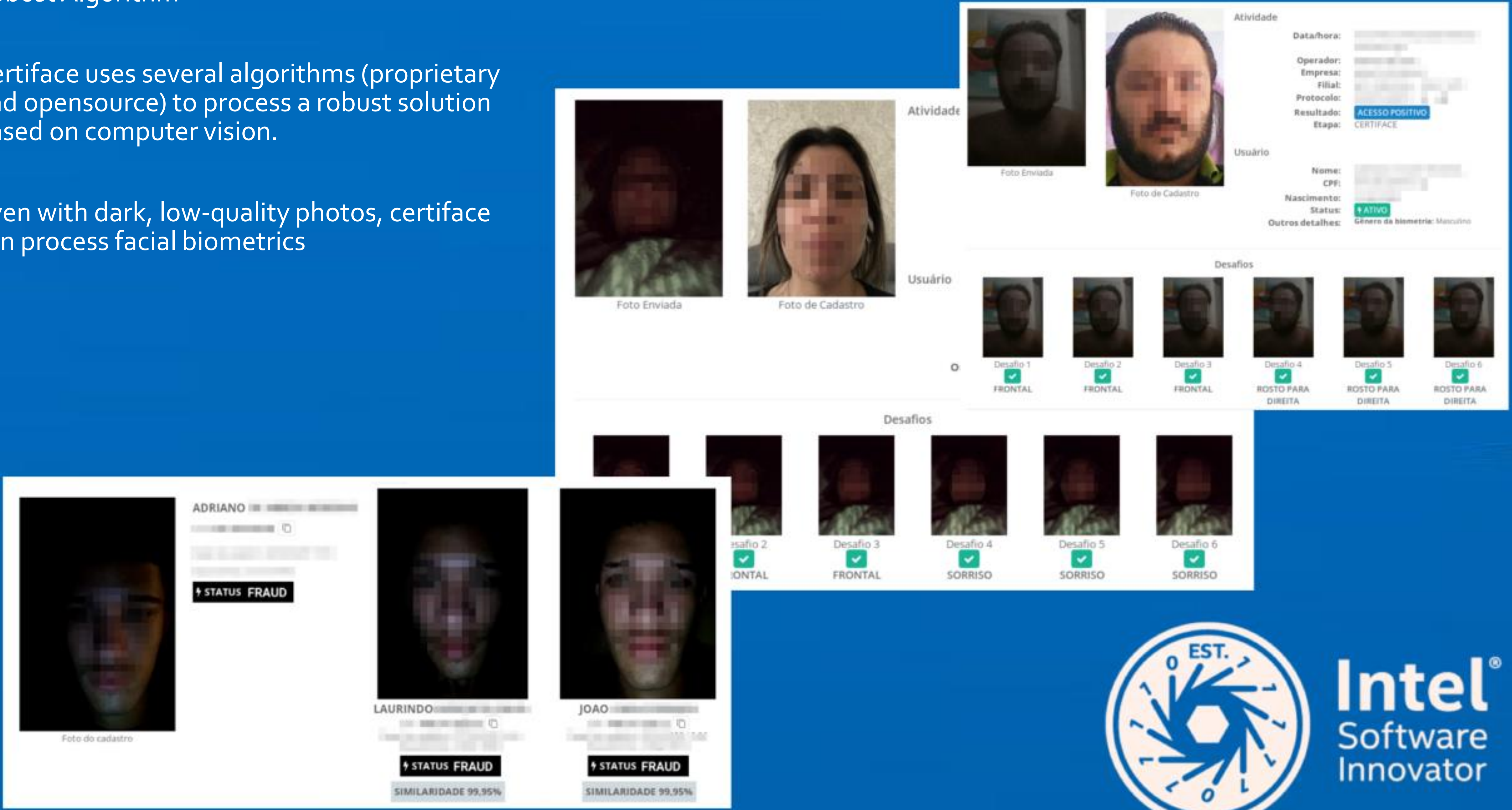


Foto Enviada

Foto de Cadastro

Atividade

Data/hora: [redacted]
Operador: [redacted]
Empresa: [redacted]
Filial: [redacted]
Protocolo: [redacted]
Resultado: ACESSO POSITIVO
Etapa: CERTIFACE

Usuário

Nome: [redacted]
CPF: [redacted]
Nascimento: [redacted]
Status: ATIVO
Outros detalhes: Gênero da biometria: Masculino

Desafios

Desafio 1 ✓ FRONTAL

Desafio 2 ✓ FRONTAL

Desafio 3 ✓ FRONTAL

Desafio 4 ✓ ROSTO PARA DIREITA

Desafio 5 ✓ ROSTO PARA DIREITA

Desafio 6 ✓ ROSTO PARA DIREITA

Foto Enviada

Foto de Cadastro

Atividade

Usuário

Desafios

esafio 2 ✓ ONTAL

Desafio 3 ✓ FRONTAL

Desafio 4 ✓ SORRISO

Desafio 5 ✓ SORRISO

Desafio 6 ✓ SORRISO

Foto do cadastro

ADRIANO [redacted]

STATUS FRAUD

LAURINDO [redacted]

STATUS FRAUD

SIMILARIDADE 99.95%

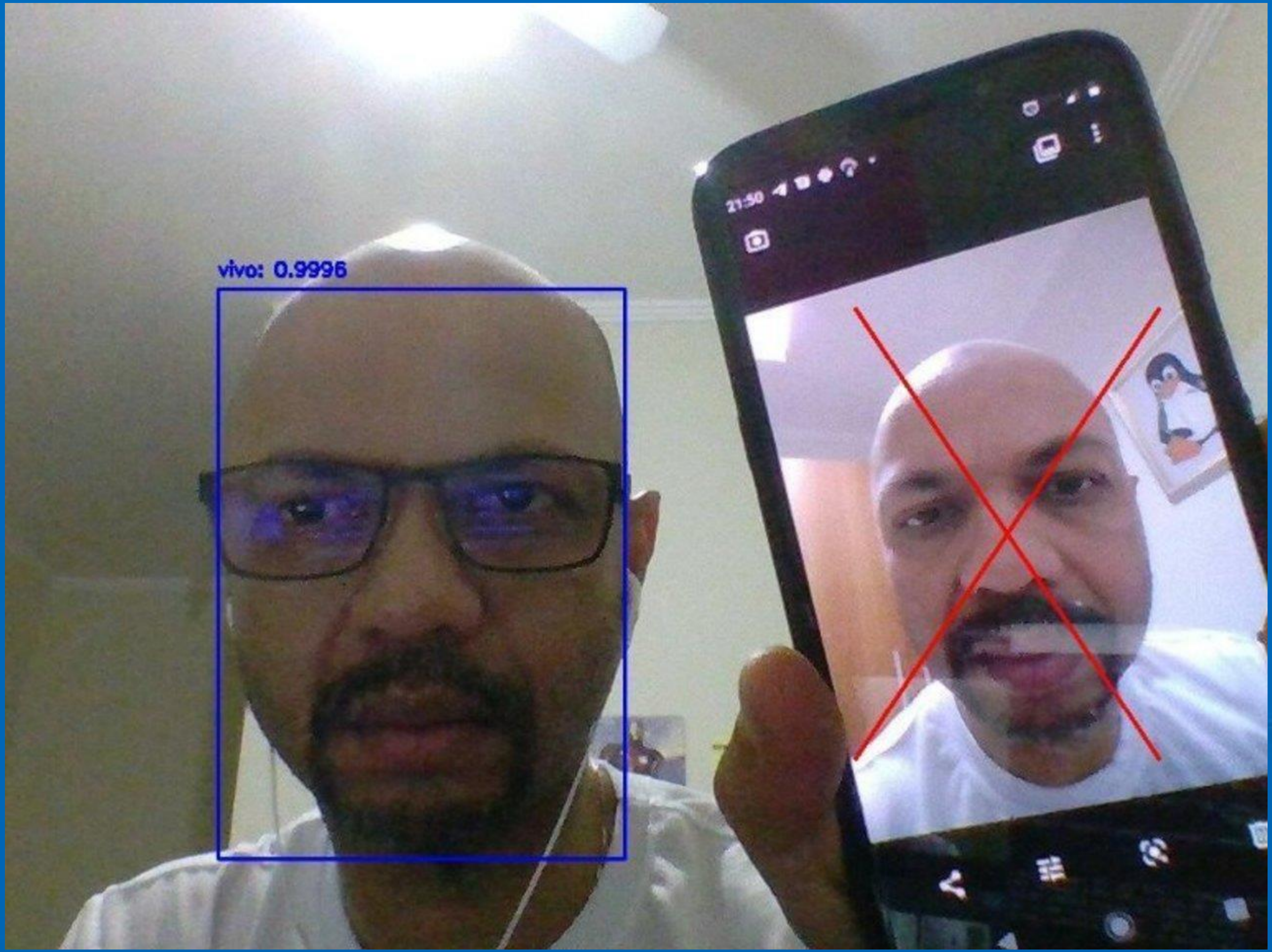
JOAO [redacted]

STATUS FRAUD

SIMILARIDADE 99.95%

EST. 2017

Intel® Software Innovator



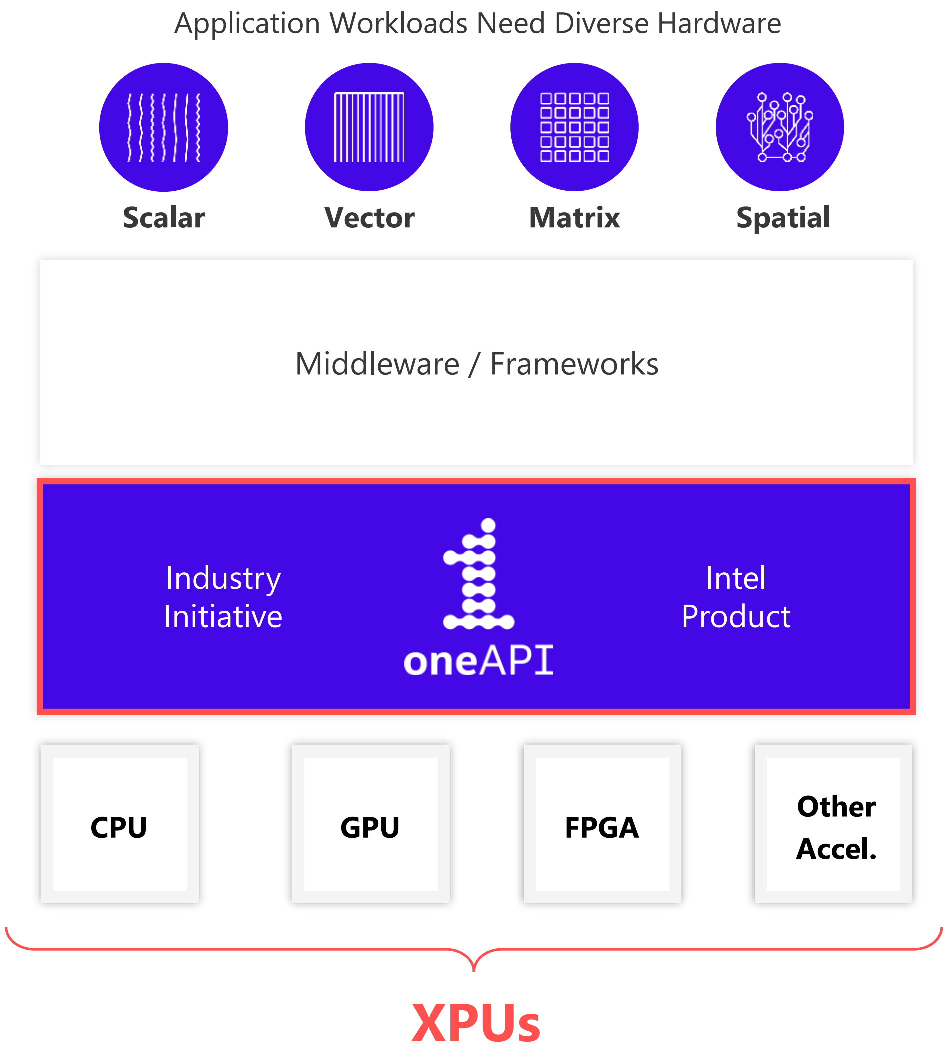
Intel®
Software
Innovator

Introducing **oneAPI**

Unified programming model to simplify development across diverse architectures

- Unified and simplified language and libraries for expressing parallelism
- Uncompromised native high-level language performance
- Based on industry standards and open specifications
- Interoperable with existing HPC programming models

Refer to http://software.intel.com/en-us/articles/optimization-notice for more information regarding performance and optimization choices in Intel software products.



oneAPI initiative – Ecosystem support

allegro.ai

CINECA

GIGASPACES

SAP

Taboola

AI SINGAPORE

Hewlett Packard Enterprise

Lenovo

Tencent 腾讯

rENIAC

Argonne NATIONAL LABORATORY

CERN openlab

codeplay

MEGWARE

PHILIPS

sas

ATLAS EXPERIMENT

CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING

IISER PUNE

Laboratório Nacional de Computação Científica

SENAI CIMATEC SISTEMA FIEB Federação das Indústrias do Estado da Bahia

SUSE We adapt. You succeed.

Ben-Gurion University of the Negev

Indian Institute of Technology Delhi

HCL

MEGH COMPUTING

Tech Mahindra

UNIVERSITY OF CAMBRIDGE

ANSYS

INDIAN INSTITUTE OF SCIENCE

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

Stockholm University

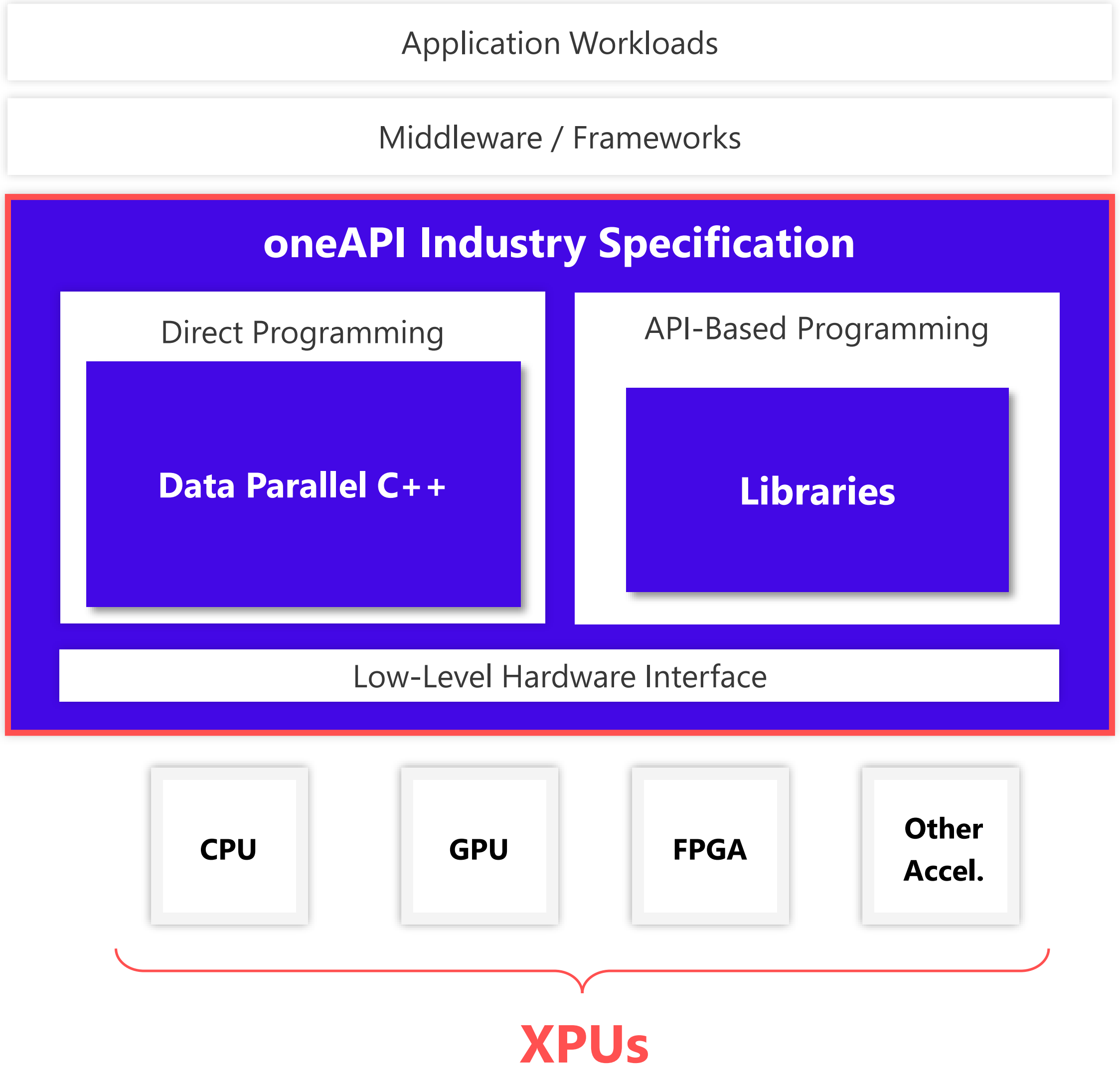
KTH VETENSKAP OCH KONST

ZUSE INSTITUTE BERLIN

These organizations support the oneAPI initiative 'concept' for a single, unified programming model for cross-architecture development. It does not indicate any agreement to purchase or use of Intel's products.

oneAPI Industry Initiative

- oneAPI Industry Specification
  - A standards based cross-architecture language, DPC++, based on C++ and SYCL
  - Powerful APIs designed for acceleration of key domain-specific functions
  - Low-level hardware interface to provide a hardware abstraction layer to vendors
  - Enables code reuse across architectures and vendors
  - Open standard to promote community and industry support
- Technical Advisory Board
- oneAPI Industry Brand



Visit oneapi.com for more details

Some capabilities may differ per architecture and custom-tuning will still be required.
Refer to http://software.intel.com/en-us/articles/optimization-notice for more information regarding performance and optimization choices in Intel software products.

Data parallel C++

Standards-based, Cross-architecture Language

Language to deliver uncompromised parallel programming productivity and performance across CPUs and accelerators

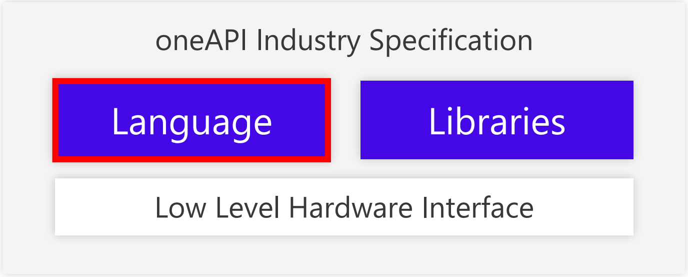
- DPC++ = ISO C++ and Khronos SYCL and Extensions
- Allows code reuse across hardware targets, while permitting custom tuning for a specific accelerator
- Open, cross-industry alternative to single architecture proprietary language

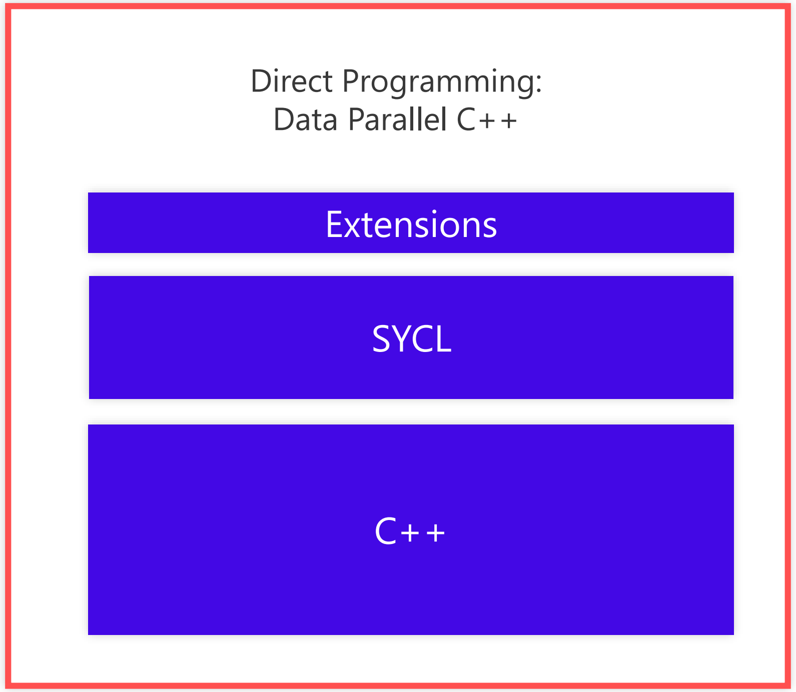
Based on C++

- Delivers C++ productivity benefits, using common and familiar C and C++ constructs
- Incorporates SYCL* from the Khronos Group to support data parallelism and heterogeneous programming

Community Project to drive language enhancements

- Extensions to simplify data parallel programming
- Open and cooperative development for continued evolution
- DPC++ extensions including Unified Shared Memory are being incorporated into upcoming versions of the Khronos SYCL standard.





Refer to http://software.intel.com/en-us/articles/optimization-notice for more information regarding performance and optimization choices in Intel software products.

Intel® oneAPI Products$^{(beta)}$

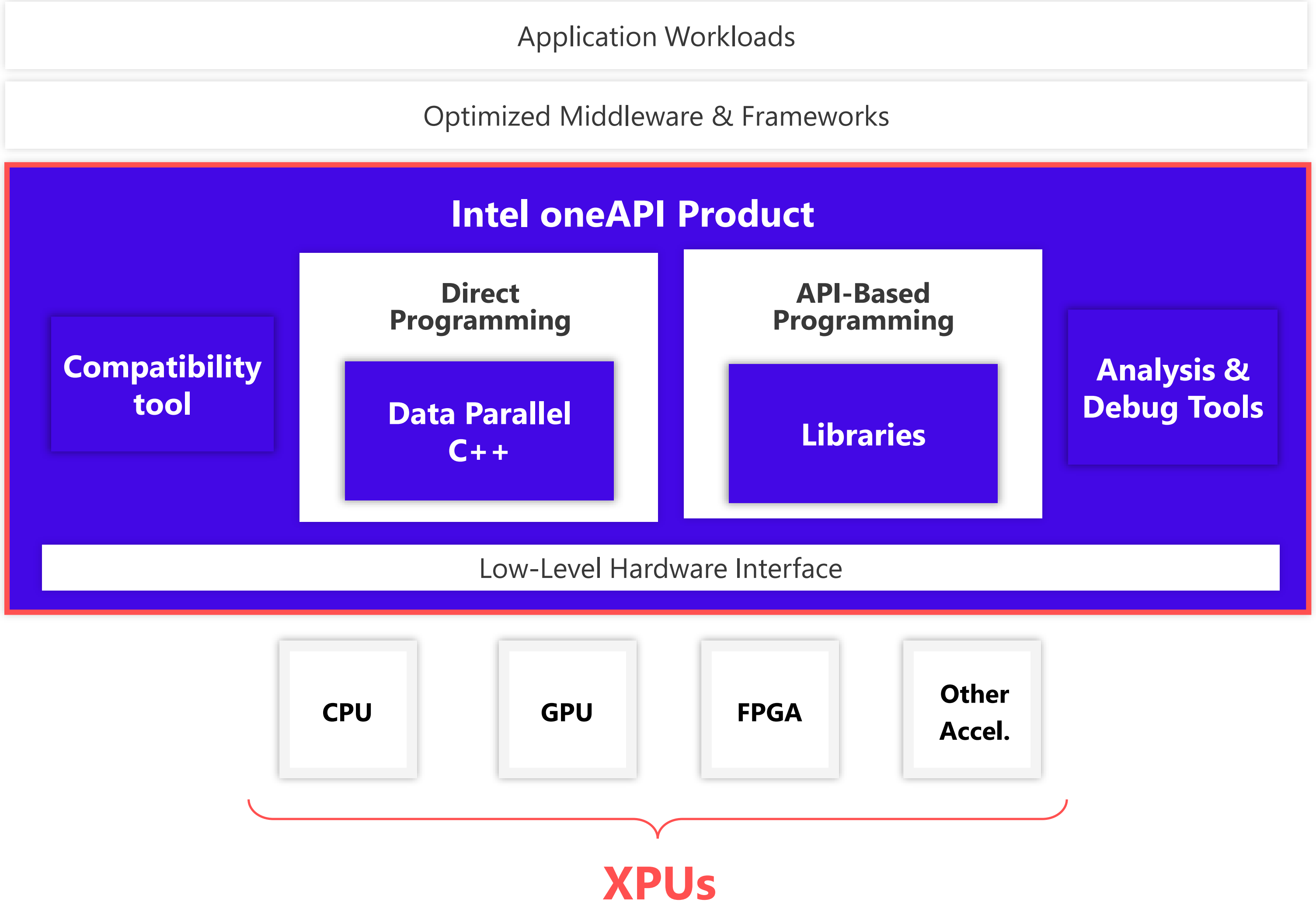
Distributed through a core toolkit and a complementary set of add-on domain-specific toolkits

Includes DPC++ compatibility tool for code migration along with advanced performance analysis and debug tools

Beta Available Now

Some capabilities may differ per architecture and custom-tuning will still be required. Other accelerators to be supported in the future.

Refer to http://software.intel.com/en-us/articles/optimization-notice for more information regarding performance and optimization choices in Intel software products.



Application Workloads

Optimized Middleware & Frameworks

Intel oneAPI Product

Compatibility tool

Direct Programming

Data Parallel C++

API-Based Programming

Libraries

Analysis & Debug Tools

Low-Level Hardware Interface

CPU

GPU

FPGA

Other Accel.

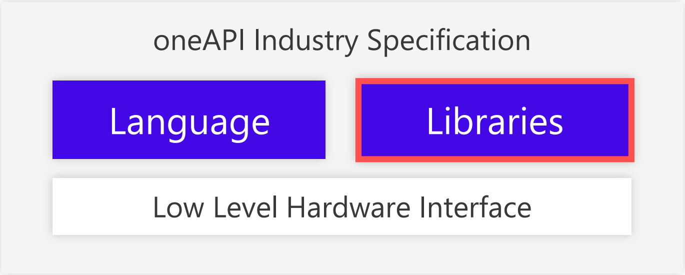
XPUs

Visit software.intel.com/oneapi for more details

oneAPI Specification Libraries

Key domain-specific functions to accelerate compute intensive workloads

Custom-coded for supported architectures



Library Name	Description	Short name
oneAPI DPC++ Library	Key algorithms and functions to speed up DPC++ kernel programming	oneDPC
oneAPI Math Kernel Library	Math routines including matrix algebra, fast Fourier transforms (FFT), and vector math	oneMKL
oneAPI Data Analytics Library	Machine learning and data analytics functions	oneDAL
oneAPI Deep Neural Network Library	Neural networks functions for deep learning training and inference	oneDNN
oneAPI Collective Communications Library	Communication patterns for distributed deep learning	oneCCL
oneAPI Threading Building Blocks	Threading and memory management template library	oneTBB
oneAPI Video Processing Library	Real-time video decoding, encoding, transcoding, and processing functions	oneVPL

Refer to http://software.intel.com/en-us/articles/optimization-notice for more information regarding performance and optimization choices in Intel software products

oneAPI Home

Intel® oneAPI Video Processing Library (Beta)

Accelerated Video Processing with a Unified Programming API

Jump to: Documentation & Code Samples | Key Specifications | Get Help

State-of-the-Art Video Codecs

The Intel® oneAPI Video Processing Library lets developers add high-speed, real-time transcoding, decoding, and encoding to their applications. Its single video API provides direct access to advanced Intel® CPU and GPU instructions, and gives you total control of the video hardware for their processing needs.

The library is perfect for applications spanning broadcasting, OTT and VOD, in-cloud gaming, and remote desktop solutions.

- Includes high-performance, hardware-accelerated AVC, HEVC, and AV1 codecs
- Supports deployment on CPUs and GPUs
- Flexible API enables developers to maximize application exposure to Intel® hardware

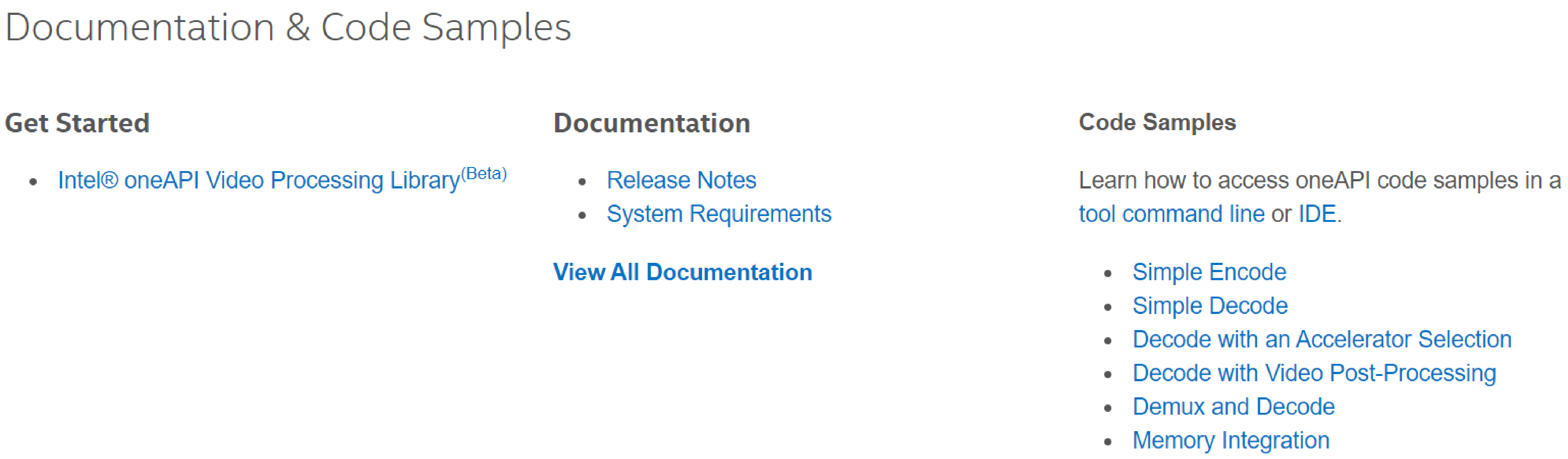
Develop, Test, and Run Your oneAPI Code in the Cloud

Get what you need to build and optimize your oneAPI projects for free. With an Intel® DevCloud account, you get 120 days of access to the latest Intel® hardware—CPUs, GPUs, FPGAs—and Intel oneAPI tools and frameworks. No software downloads. No configuration steps. No installations.

Get Access

Download Intel oneAPI Video Processing Library as Part of the Intel® oneAPI Base Toolkit

Get It Now



Documentation & Code Samples

Get Started

- Intel® oneAPI Video Processing Library(Beta)

Documentation

- Release Notes
- System Requirements

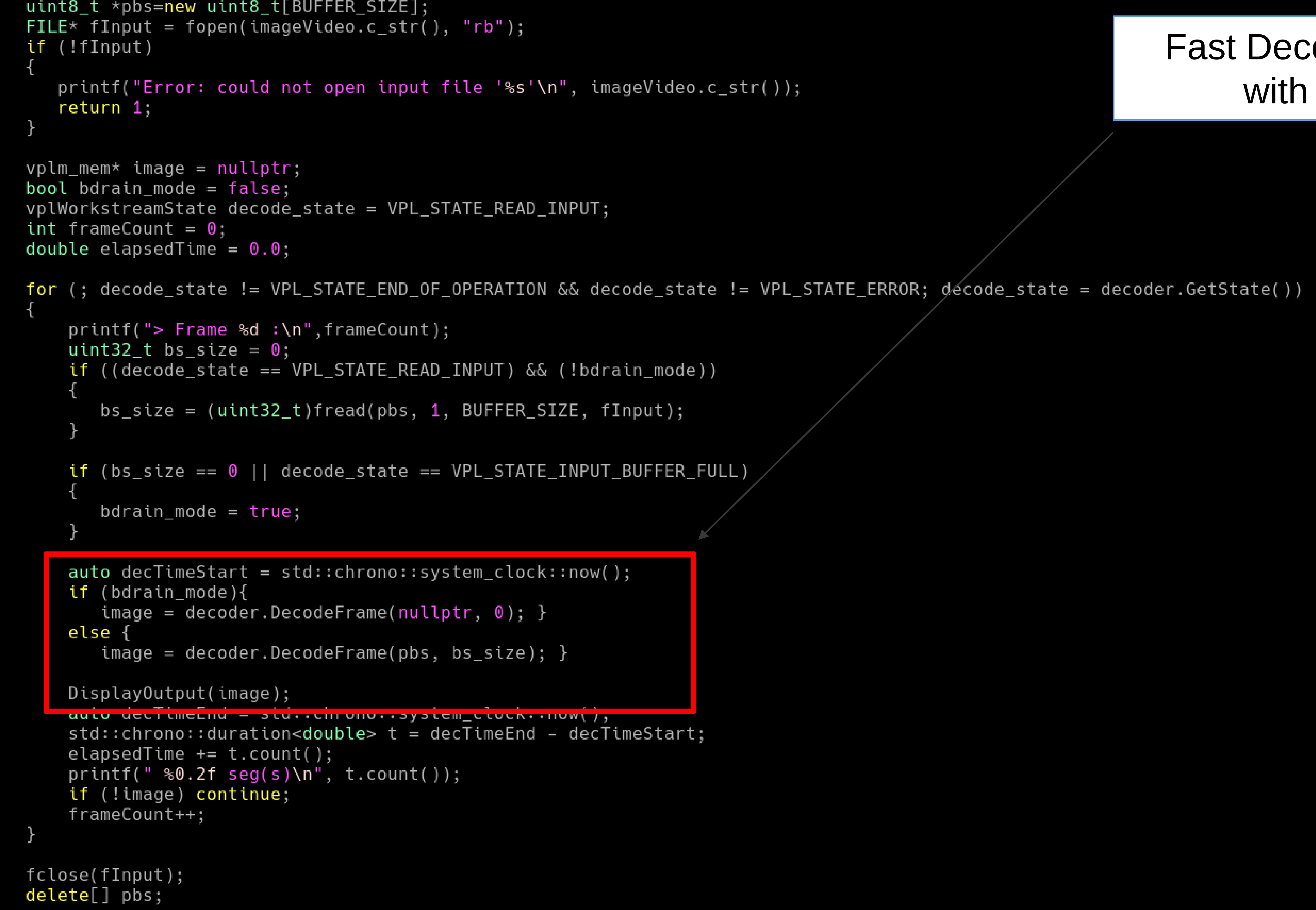
View All Documentation

Code Samples

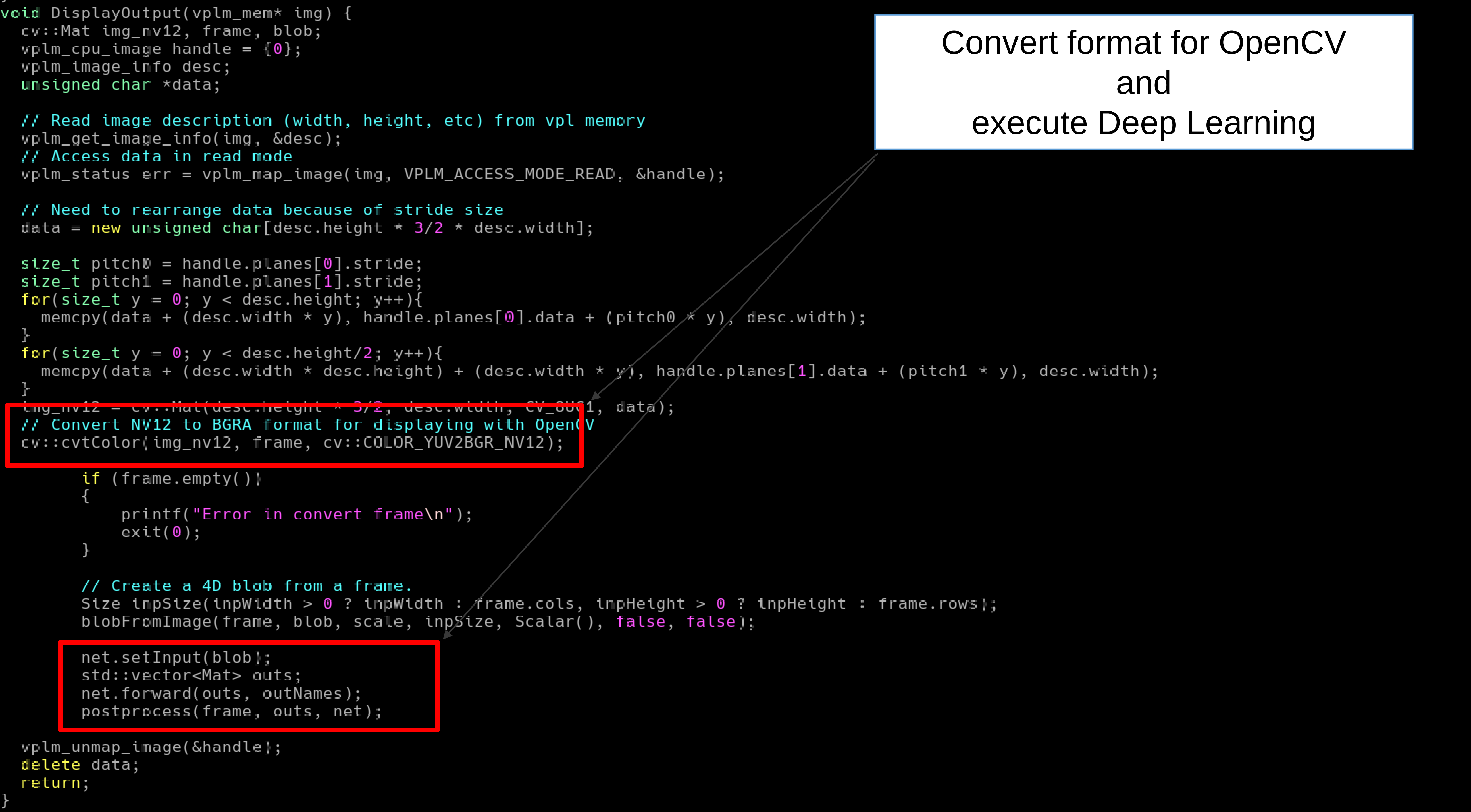
Learn how to access oneAPI code samples in a tool command line or IDE.

- Simple Encode
- Simple Decode
- Decode with an Accelerator Selection
- Decode with Video Post-Processing
- Demux and Decode
- Memory Integration

Fast Decoder in C++ with libVPL

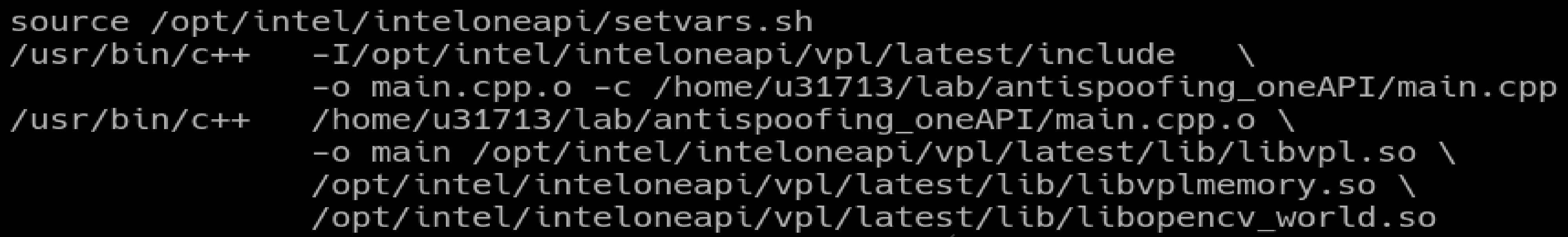


```
uint8_t *pbs=new uint8_t[BUFFER_SIZE];
FILE* fInput = fopen(imageVideo.c_str(), "rb");
if (!fInput)
{
   printf("Error: could not open input file '%s'\n", imageVideo.c_str());
   return 1;
}

vplm_mem* image = nullptr;
bool bdrain_mode = false;
vplWorkstreamState decode_state = VPL_STATE_READ_INPUT;
int frameCount = 0;
double elapsedTime = 0.0;

for (; decode_state != VPL_STATE_END_OF_OPERATION && decode_state != VPL_STATE_ERROR; decode_state = decoder.GetState())
{
    printf("> Frame %d :\n",frameCount);
    uint32_t bs_size = 0;
    if ((decode_state == VPL_STATE_READ_INPUT) && (!bdrain_mode))
    {
       bs_size = (uint32_t)fread(pbs, 1, BUFFER_SIZE, fInput);
    }

    if (bs_size == 0 || decode_state == VPL_STATE_INPUT_BUFFER_FULL)
    {
       bdrain_mode = true;
    }

    auto decTimeStart = std::chrono::system_clock::now();
    if (bdrain_mode){
       image = decoder.DecodeFrame(nullptr, 0); }
    else {
       image = decoder.DecodeFrame(pbs, bs_size); }

    DisplayOutput(image);
    auto decTimeEnd = std::chrono::system_clock::now();
    std::chrono::duration<double> t = decTimeEnd - decTimeStart;
    elapsedTime += t.count();
    printf(" %0.2f seg(s)\n", t.count());
    if (!image) continue;
    frameCount++;
}

fclose(fInput);
delete[] pbs;
```



```
void DisplayOutput(vplm_mem* img) {
  cv::Mat img_nv12, frame, blob;
  vplm_cpu_image handle = {0};
  vplm_image_info desc;
  unsigned char *data;

  // Read image description (width, height, etc) from vpl memory
  vplm_get_image_info(img, &desc);
  // Access data in read mode
  vplm_status err = vplm_map_image(img, VPLM_ACCESS_MODE_READ, &handle);

  // Need to rearrange data because of stride size
  data = new unsigned char[desc.height * 3/2 * desc.width];

  size_t pitch0 = handle.planes[0].stride;
  size_t pitch1 = handle.planes[1].stride;
  for(size_t y = 0; y < desc.height; y++){
    memcpy(data + (desc.width * y), handle.planes[0].data + (pitch0 * y), desc.width);
  }
  for(size_t y = 0; y < desc.height/2; y++){
    memcpy(data + (desc.width * desc.height) + (desc.width * y), handle.planes[1].data + (pitch1 * y), desc.width);
  }
  img_nv12 = cv::Mat(desc.height * 3/2, desc.width, CV_8UC1, data);
  // Convert NV12 to BGRA format for displaying with OpenCV
  cv::cvtColor(img_nv12, frame, cv::COLOR_YUV2BGR_NV12);

        if (frame.empty())
        {
            printf("Error in convert frame\n");
            exit(0);
        }

        // Create a 4D blob from a frame.
        Size inpSize(inpWidth > 0 ? inpWidth : frame.cols, inpHeight > 0 ? inpHeight : frame.rows);
        blobFromImage(frame, blob, scale, inpSize, Scalar(), false, false);

        net.setInput(blob);
        std::vector<Mat> outs;
        net.forward(outs, outNames);
        postprocess(frame, outs, net);

  vplm_unmap_image(&handle);
  delete data;
  return;
}
```

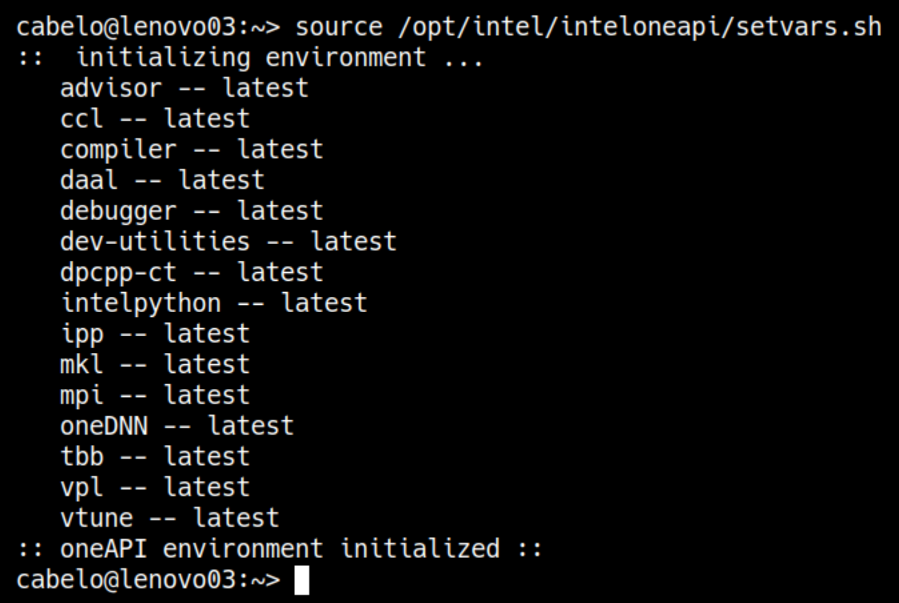
Convert format for OpenCV
and
execute Deep Learning



```
source /opt/intel/inteloneapi/setvars.sh
/usr/bin/c++   -I/opt/intel/inteloneapi/vpl/latest/include   \
               -o main.cpp.o -c /home/u31713/lab/antispoofing_oneAPI/main.cpp
/usr/bin/c++   /home/u31713/lab/antispoofing_oneAPI/main.cpp.o \
               -o main /opt/intel/inteloneapi/vpl/latest/lib/libvpl.so \
               /opt/intel/inteloneapi/vpl/latest/lib/libvplmemory.so \
               /opt/intel/inteloneapi/vpl/latest/lib/libopencv_world.so
```

How to Build

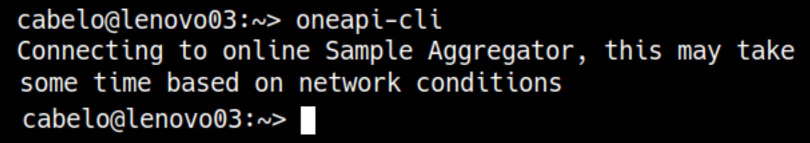
- I take Linux seriously, researches and works with biometrics and computer vision since 1998.
- SUSE and openSUSE Leap Linux distribution
- openCV optimized for Intel processors
- Based on the Simple Decode example



```
cabelo@lenovo03:~> source /opt/intel/inteloneapi/setvars.sh
::  initializing environment ...
   advisor -- latest
   ccl -- latest
   compiler -- latest
   daal -- latest
   debugger -- latest
   dev-utilities -- latest
   dpcpp-ct -- latest
   intelpython -- latest
   ipp -- latest
   mkl -- latest
   mpi -- latest
   oneDNN -- latest
   tbb -- latest
   vpl -- latest
   vtune -- latest
:: oneAPI environment initialized ::
cabelo@lenovo03:~>
```

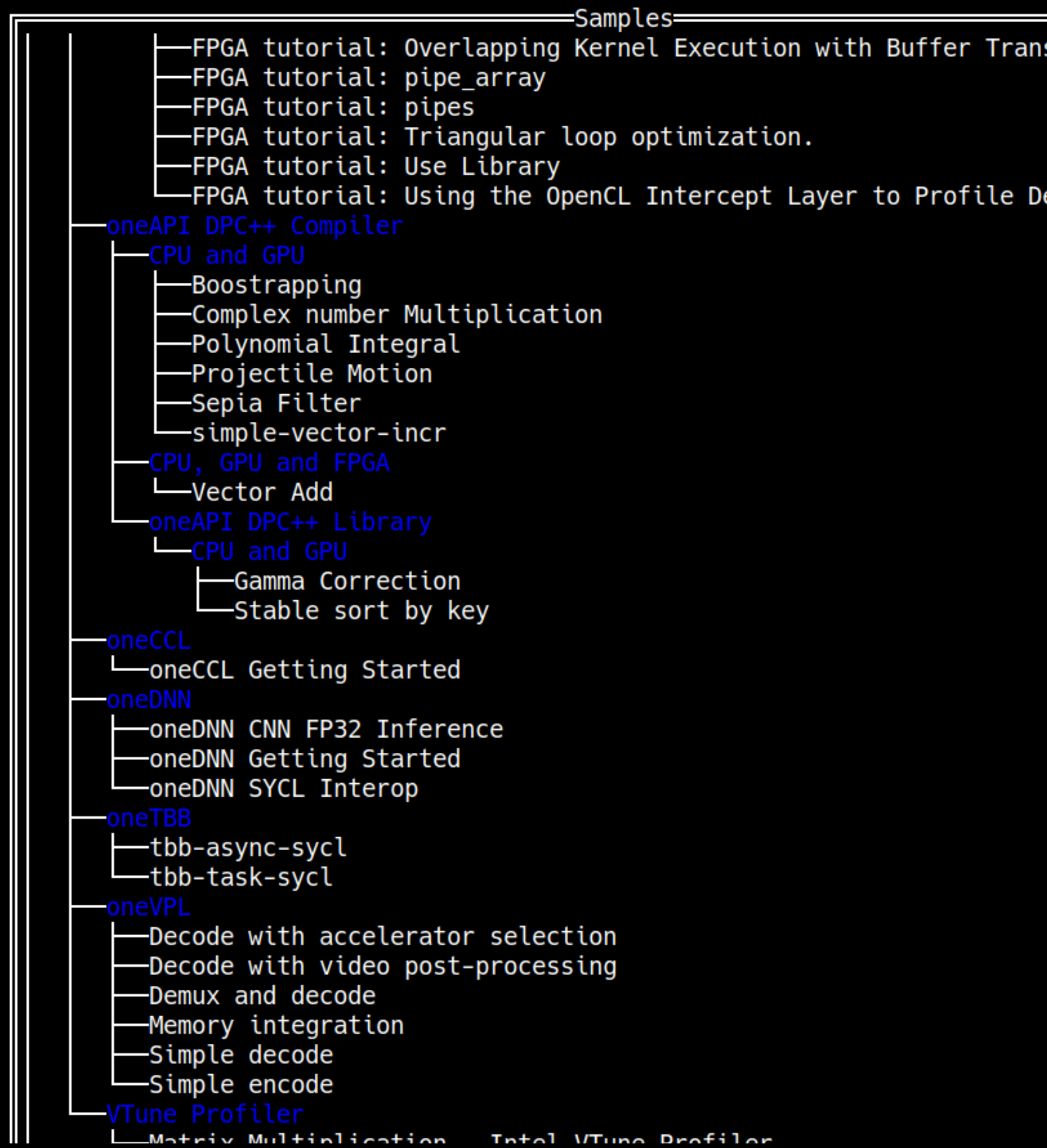
Environment Setting

Menu Script

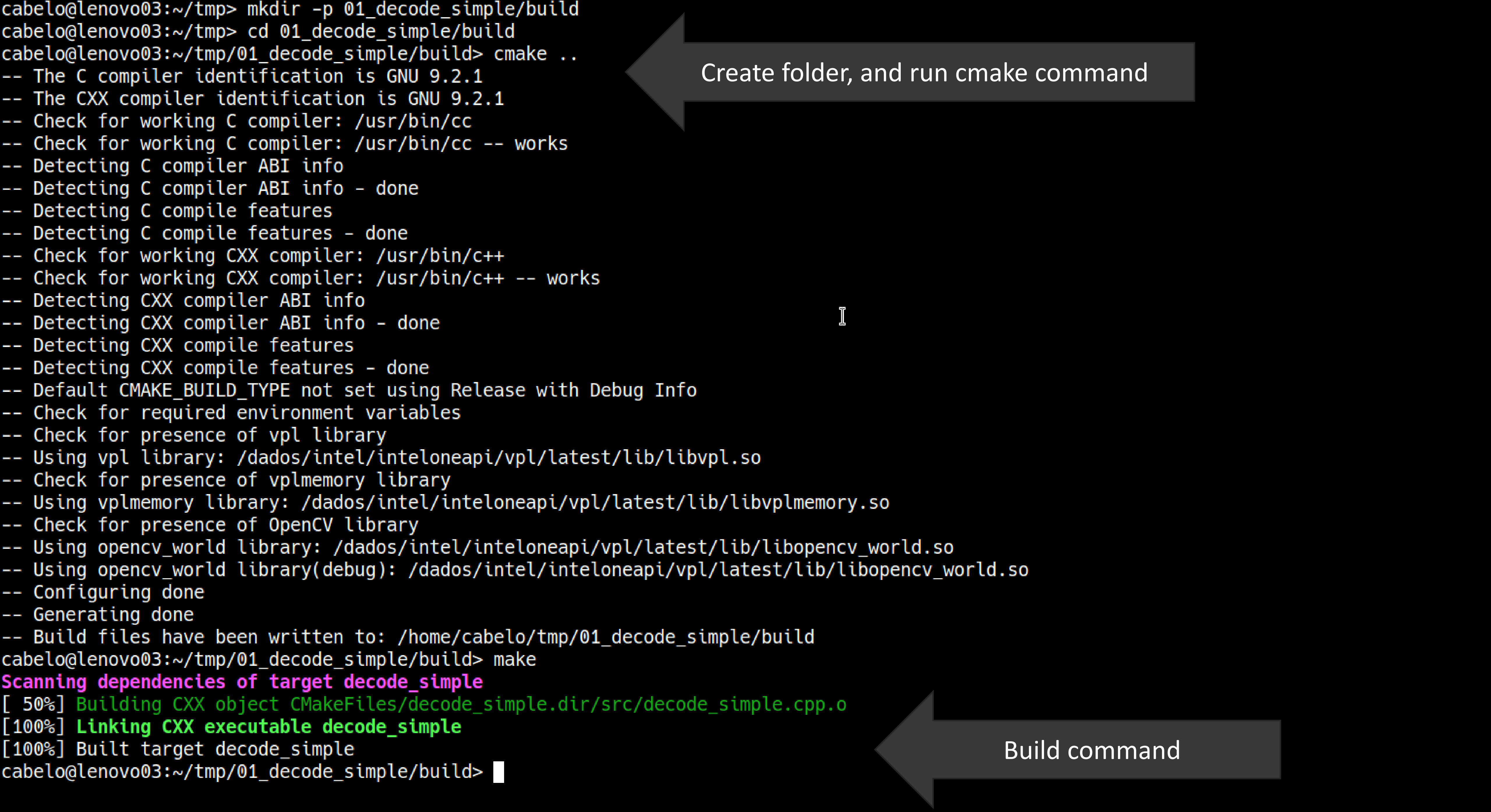


```
cabelo@lenovo03:~> oneapi-cli
Connecting to online Sample Aggregator, this may take
some time based on network conditions
cabelo@lenovo03:~>
```

Samples



```
╔═══════════════════════════════════════Samples══════════
║│ │   ├─FPGA tutorial: Overlapping Kernel Execution with Buffer Trans
║│ │   ├─FPGA tutorial: pipe_array
║│ │   ├─FPGA tutorial: pipes
║│ │   ├─FPGA tutorial: Triangular loop optimization.
║│ │   ├─FPGA tutorial: Use Library
║│ │   └─FPGA tutorial: Using the OpenCL Intercept Layer to Profile De
║│ ├─oneAPI DPC++ Compiler
║│ │ ├─CPU and GPU
║│ │ │ ├─Boostrapping
║│ │ │ ├─Complex number Multiplication
║│ │ │ ├─Polynomial Integral
║│ │ │ ├─Projectile Motion
║│ │ │ ├─Sepia Filter
║│ │ │ └─simple-vector-incr
║│ │ ├─CPU, GPU and FPGA
║│ │ │ └─Vector Add
║│ │ └─oneAPI DPC++ Library
║│ │   └─CPU and GPU
║│ │     ├─Gamma Correction
║│ │     └─Stable sort by key
║│ ├─oneCCL
║│ │ └─oneCCL Getting Started
║│ ├─oneDNN
║│ │ ├─oneDNN CNN FP32 Inference
║│ │ ├─oneDNN Getting Started
║│ │ └─oneDNN SYCL Interop
║│ ├─oneTBB
║│ │ ├─tbb-async-sycl
║│ │ └─tbb-task-sycl
║│ ├─oneVPL
║│ │ ├─Decode with accelerator selection
║│ │ ├─Decode with video post-processing
║│ │ ├─Demux and decode
║│ │ ├─Memory integration
║│ │ ├─Simple decode
║│ │ └─Simple encode
║│ └─VTune Profiler
║│   └─Matrix Multiplication - Intel VTune Profiler
```



```
cabelo@lenovo03:~/tmp> mkdir -p 01_decode_simple/build
cabelo@lenovo03:~/tmp> cd 01_decode_simple/build
cabelo@lenovo03:~/tmp/01_decode_simple/build> cmake ..
-- The C compiler identification is GNU 9.2.1
-- The CXX compiler identification is GNU 9.2.1
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Default CMAKE_BUILD_TYPE not set using Release with Debug Info
-- Check for required environment variables
-- Check for presence of vpl library
-- Using vpl library: /dados/intel/inteloneapi/vpl/latest/lib/libvpl.so
-- Check for presence of vplmemory library
-- Using vplmemory library: /dados/intel/inteloneapi/vpl/latest/lib/libvplmemory.so
-- Check for presence of OpenCV library
-- Using opencv_world library: /dados/intel/inteloneapi/vpl/latest/lib/libopencv_world.so
-- Using opencv_world library(debug): /dados/intel/inteloneapi/vpl/latest/lib/libopencv_world.so
-- Configuring done
-- Generating done
-- Build files have been written to: /home/cabelo/tmp/01_decode_simple/build
cabelo@lenovo03:~/tmp/01_decode_simple/build> make
Scanning dependencies of target decode_simple
[ 50%] Building CXX object CMakeFiles/decode_simple.dir/src/decode_simple.cpp.o
[100%] Linking CXX executable decode_simple
[100%] Built target decode_simple
cabelo@lenovo03:~/tmp/01_decode_simple/build>
```

Create folder, and run cmake command

Build command

Create H.264 DECODER

```
vpl::Workstream decoder(VPL_TARGET_DEVICE_DEFAULT, VPL_WORKSTREAM_DECODE);
```

Use GPU if available

```
decoder.SetConfig(VPL_PROP_SRC_BITSTREAM_FORMAT, VPL_FOURCC_H264);
```

How To Work

```
vplm_mem* image = nullptr;
```

Read FRAME

```
stream.read(reinterpret_cast(buffer.data()), buffer.size());
```

Decoder frame

```
image = decoder.DecodeFrame(buffer.data(), stream.gcount());
```

Access data in read mode

```
vplm_status err = vplm_map_image(image, VPLM_ACCESS_MODE_READ, &handle);
```

Summary

Certiface technology was designed to be used in Banking system to combat fraud and protect honest people by using technology to differentiate between a living person and a recorded video

Certiface AntiSpoofing use oneAPI and one VPL for fast decode video

oneVPL Library enables allows the development of real-time transcoding, decoding and encoding, high-speed direct access to advanced Intel CPU and GPU instructions

oneAPI unifies and simplifies programming across CPUs and accelerators, delivering developer productivity across architectures and vendors.

INNOVATOR PROGRAM: RECOGNIZING YOUR WORK IN THE COMMUNITY (DEVMESH.INTEL.COM-> MEMBER PROGRAMS)

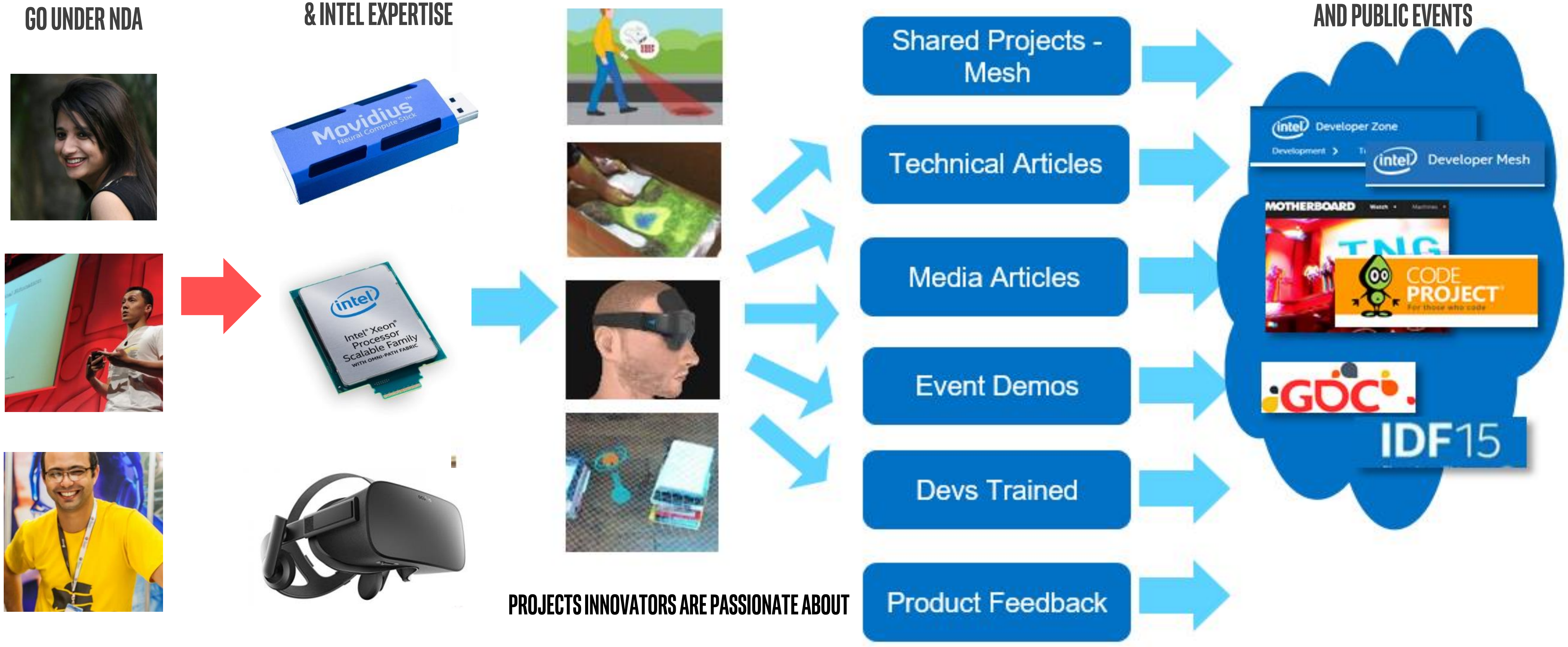
TOP TIER EXPERTS WHO INNOVATE & LIKE TO SHARE WORK GO UNDER NDA

ACCESS TO INTEL HW/SW & INTEL EXPERTISE

EARLY EXPERIMENTS, PROTOTYPES, TESTING & RESEARCH

INNOVATOR SPOTLIGHT

INNOVATOR SPEAKERSHIPS & DEMOS AT MAJOR CONFERENCES AND PUBLIC EVENTS



Shared Projects - Mesh

Technical Articles

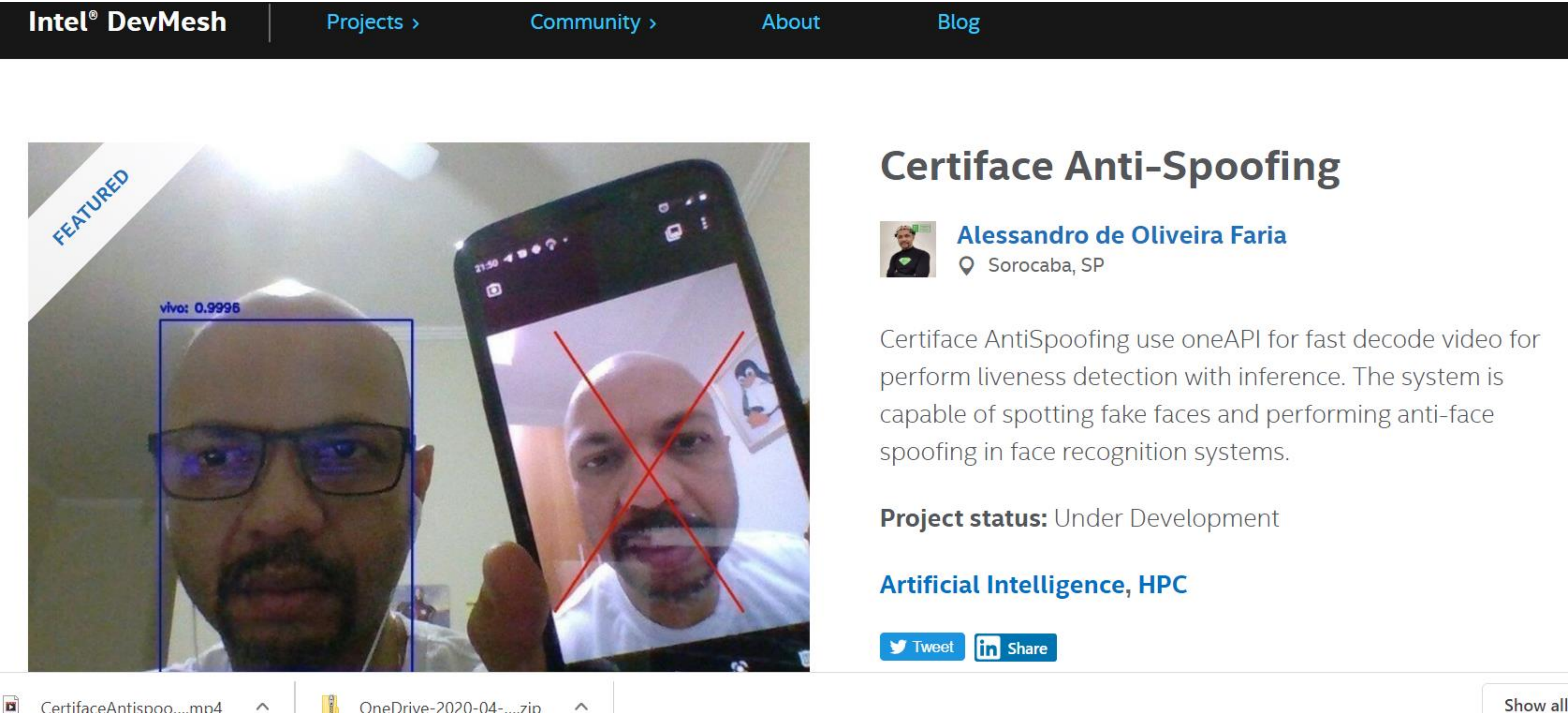
Media Articles

Event Demos

Devs Trained

Product Feedback

PROJECTS INNOVATORS ARE PASSIONATE ABOUT



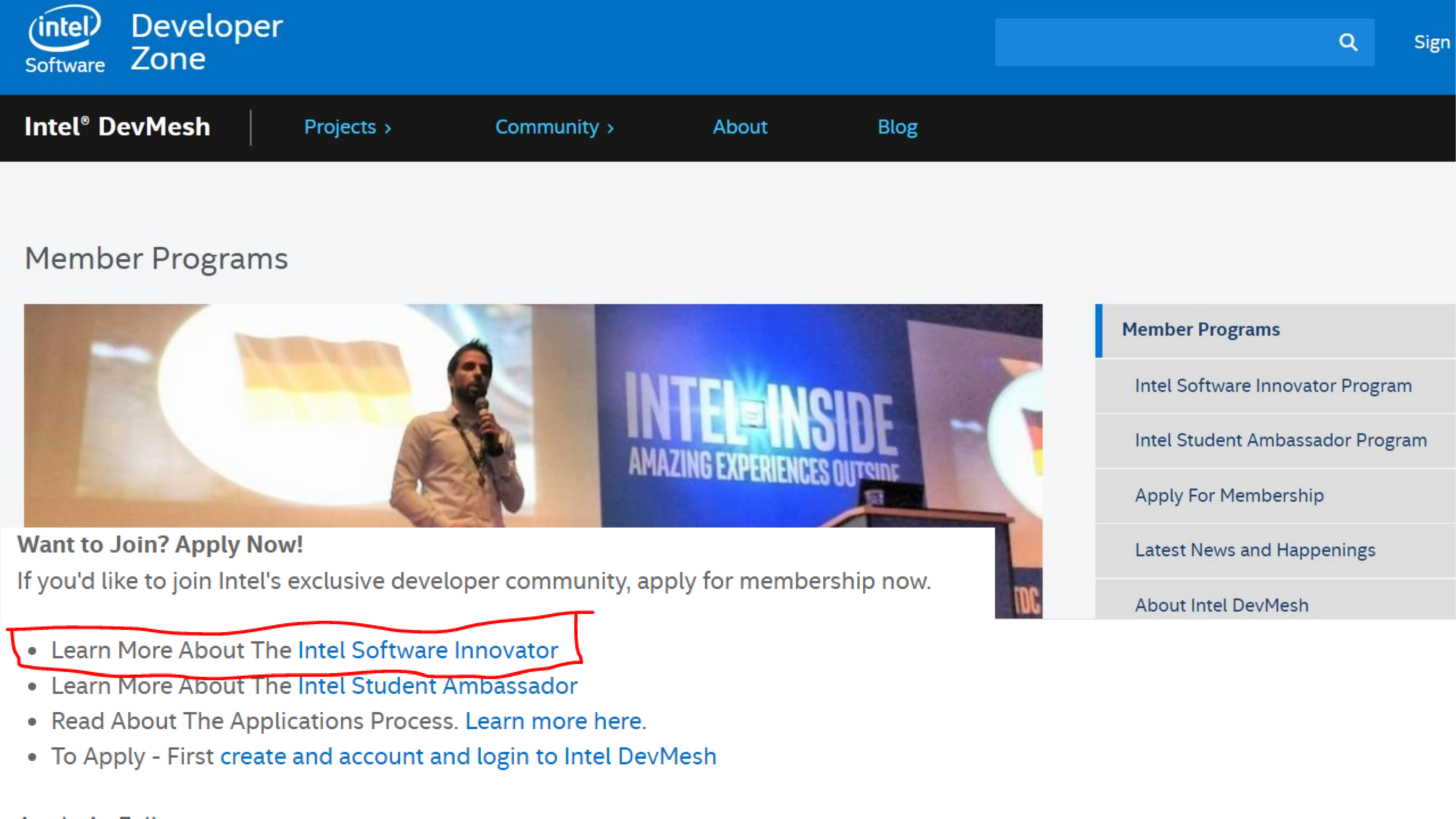
Certiface Anti-Spoofing

Alessandro de Oliveira Faria
Sorocaba, SP

Certiface AntiSpoofing use oneAPI for fast decode video for perform liveness detection with inference. The system is capable of spotting fake faces and performing anti-face spoofing in face recognition systems.

Project status: Under Development

Artificial Intelligence, HPC

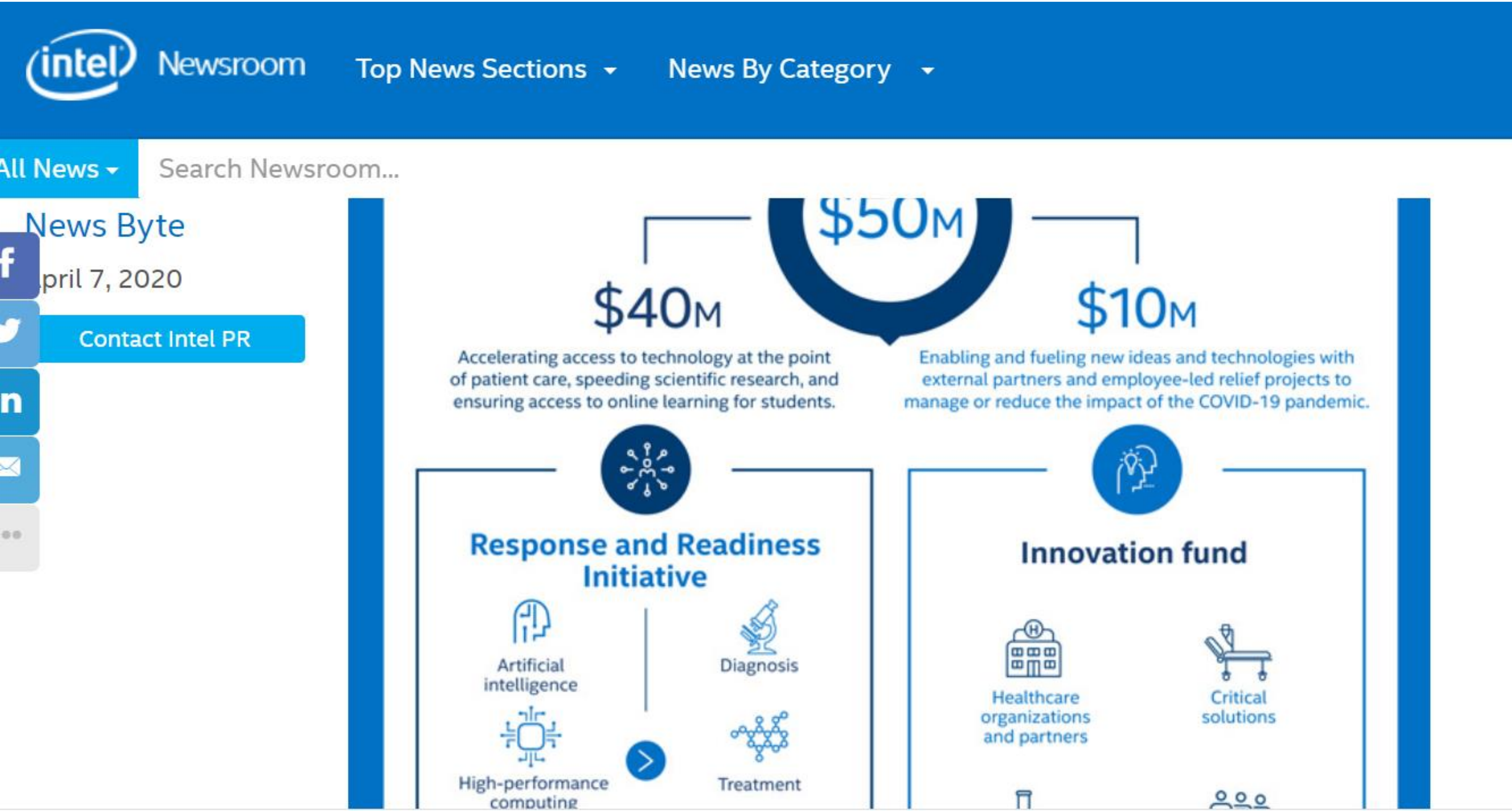


Member Programs

Want to Join? Apply Now!

If you'd like to join Intel's exclusive developer community, apply for membership now.

- Learn More About The Intel Software Innovator
- Learn More About The Intel Student Ambassador
- Read About The Applications Process. Learn more here.
- To Apply - First create and account and login to Intel DevMesh



News Byte

pril 7, 2020

$50M

$40M

Accelerating access to technology at the point of patient care, speeding scientific research, and ensuring access to online learning for students.

$10M

Enabling and fueling new ideas and technologies with external partners and employee-led relief projects to manage or reduce the impact of the COVID-19 pandemic.

Response and Readiness Initiative

Innovation fund

THANK YOU

Alessandro de Oliveira Faria
Sujata Tibrewala
SUJATA.TIBREWALA@INTEL.COM
@SUJATATIBRE