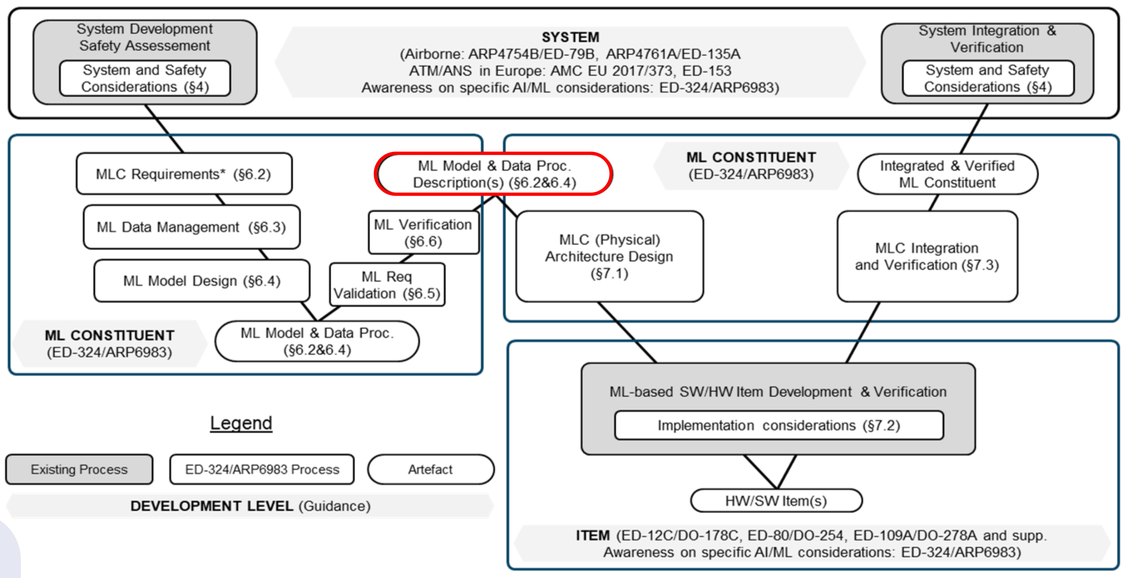
The SONNX Working Group

Towards a Safe ONNX Profile

Eric JENN, Mariem TURKI, IRT Saint Exupery, *et al*

ML models and Certification

- The ML model play a pivotal role in the development process
- Applicants will have to provide evidences to support safety demonstration and certification



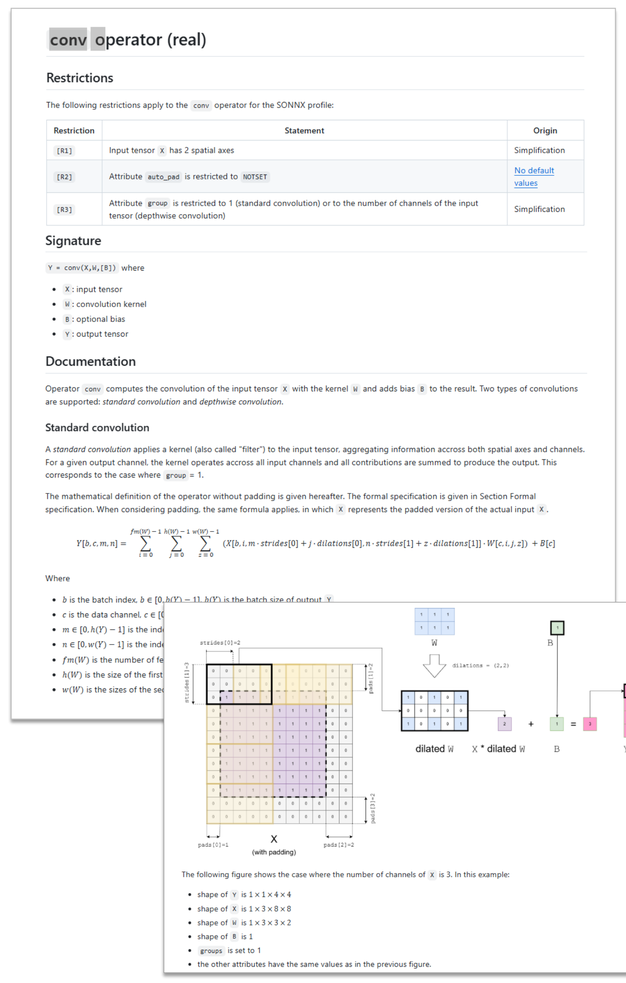
From ONNX to SONNX…

- ONNX is the de facto standard for ML model exchange
- ONNX has not been developed with safety in mind
- An ONNX model remain interpretable, leading to different behaviours depending on the ML implementation framework
- ONNX needs to be clarified, verified, completed, corrected….

The SONNX profile

https://github.com/ericjenn/working-groups/blob/ericjenn-srpwg-wg1/safety-related-profile

- A set of operators
- A set of restrictions
- A set of specifications
- A reference implementation



conv operator (real)

Restrictions

The following restrictions apply to the `conv` operator for the SONNX profile:

Restriction	Statement	Origin
`[R1]`	Input tensor `X` has 2 spatial axes	Simplification
`[R2]`	Attribute `auto_pad` is restricted to `NOTSET`	No default values
`[R3]`	Attribute `group` is restricted to 1 (standard convolution) or to the number of channels of the input tensor (depthwise convolution)	Simplification

Signature

`Y = conv(X,W,[B])` where

- `X` : input tensor
- `W` : convolution kernel
- `B` : optional bias
- `Y` : output tensor

Documentation

Operator `conv` computes the convolution of the input tensor `X` with the kernel `W` and adds bias `B` to the result. Two types of convolutions are supported: *standard convolution* and *depthwise convolution*.

Standard convolution

A *standard convolution* applies a kernel (also called "filter") to the input tensor, aggregating information accross both spatial axes and channels. For a given output channel, the kernel operates accross all input channels and all contributions are summed to produce the output. This corresponds to the case where `group` = 1.

The mathematical definition of the operator without padding is given hereafter. The formal specification is given in Section Formal specification. When considering padding, the same formula applies, in which `X` represents the padded version of the actual input `X`.

$$Y[b,c,m,n] = \sum_{i=0}^{fm(W)-1} \sum_{j=0}^{h(W)-1} \sum_{z=0}^{w(W)-1} (X[b,i,m \cdot strides[0] + j \cdot dilations[0], n \cdot strides[1] + z \cdot dilations[1]] \cdot W[c,i,j,z]) + B[c]$$

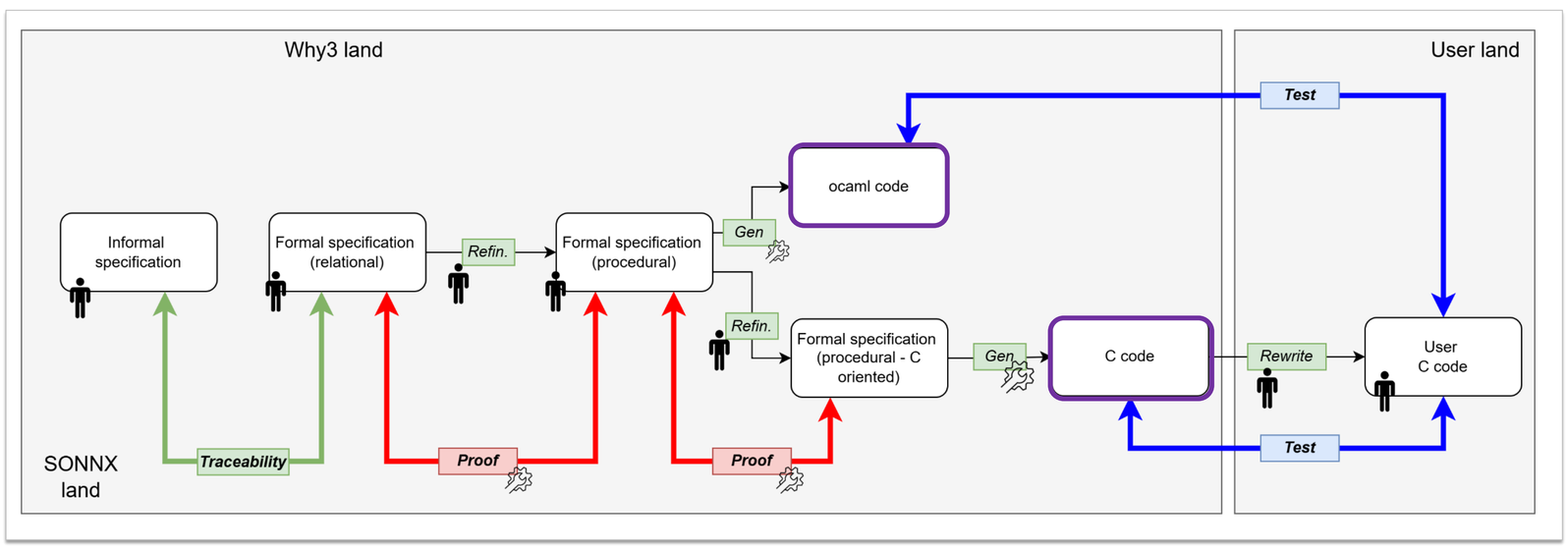
Where

- b is the batch index, $b \in [0, h(Y) - 1]$, $h(Y)$ is the batch size of output `Y`
- c is the data channel, $c \in [0$
- $m \in [0, h(Y) - 1]$ is the ind
- $n \in [0, w(Y) - 1]$ is the ind
- $fm(W)$ is the number of fe
- $h(W)$ is the size of the first
- $w(W)$ is the sizes of the se

The following figure shows the case where the number of channels of `X` is 3. In this example:

- shape of `Y` is 1 × 1 × 4 × 4
- shape of `X` is 1 × 3 × 8 × 8
- shape of `W` is 1 × 3 × 3 × 2
- shape of `B` is 1
- `groups` is set to 1
- the other attributes have the same values as in the previous figure.

From Informal to Formal specification



- Provision of an informal and formal specification of ONNX operators
- Tracability between informal and formal specification
- Formal verification of formal specification
- Generation of a reference implementation of operators

Let a be the concatenation axis and $d_{k,a}$ **(T2)** the dimension of [illegible] input tensor k along the axis a.

Let s_k be the cumulative offset along axis before input X_k as:

$$\text{T2: } s_k = \sum_{j=0}^{k-1} d_{j,a}$$

Let i_a be the global index along dimension a, and let $i_a^{'}$ be the corresponding local index within a local tensor X_k. This relationship can be defined as follows:

$$\text{T3: } i_a^{'} = i_a - s_k$$

If the global index i_a satisfies the condition:

$$\text{T4: } s_k \leq i_a < s_k + d_{k,a}$$

then the relationship holds:

$$\text{T5: } \forall i_0, \ldots, i_{r-1} \,.\, Y[i_0, \ldots, i_{r-1}] = X_k[i_0, \ldots, i_a^{'}, \ldots, i_{r-1}]$$

With i_0 and i_{r-1} are the indices which access respectively the first and last dimensio[illegible] **r-dimensional** tensor. $i_0, \ldots, i_{r-1}$ represent a set of indices that uniquely identify an element within an **r-dimensional** tensor.

```
(** T3: defining the l[illegible]dex i' for a i (global index) given **)
(*
  seq_D_axis: seq_D_axis is the sequence of all the d_k,axis for each k tensor
  i_axis: value on the axis along which the concatenation is performed
  k: Current index in sequence seq_D_axis. Must be 0 at the first call of the
  function
  s_k: Current sum of the previous dimensions in seq_D_axis. Must be 0 at the
  first call of the function
*)
let rec rec_find_k_and_i_prime (seq_D_axis: seq int) (i_axis: int) (k: int)
(s_k: int) : (int, int)
 variant { length seq_D_axis - k }  (* Termination measure *)
=
  if i_axis < s_k + seq_D_axis[k] then (* Inequality (T5) to define to in which
  k tensor the local index i' is defined *)
    (k, (i_axis - s_k)) (* Definition (T5) by keeping the upper part of the
    inequality: i' = i - s_k *)
  else
    (* The global index i is superior to s_k + seq_D_axis[k] so the k tensor to
    define the local index i' is not the current one but next one k+1. Update
    the offset (s_k) by adding the length of the tensor we just checked
    (current tensor). *)
    rec_find_k_and_i_prime seq_D_axis i_axis (k + 1) (s_k + seq_D_axis[k])
```

Validating implementations using Aidge

- Aidge is an open-source and collaborative platform to ease the optimization and the deployment of embedded AI.
- Integration of SONNX operators in Aidge
- Generation of a SONNX-compliant reference implementation to serve as a test oracle for the validation of an implementation