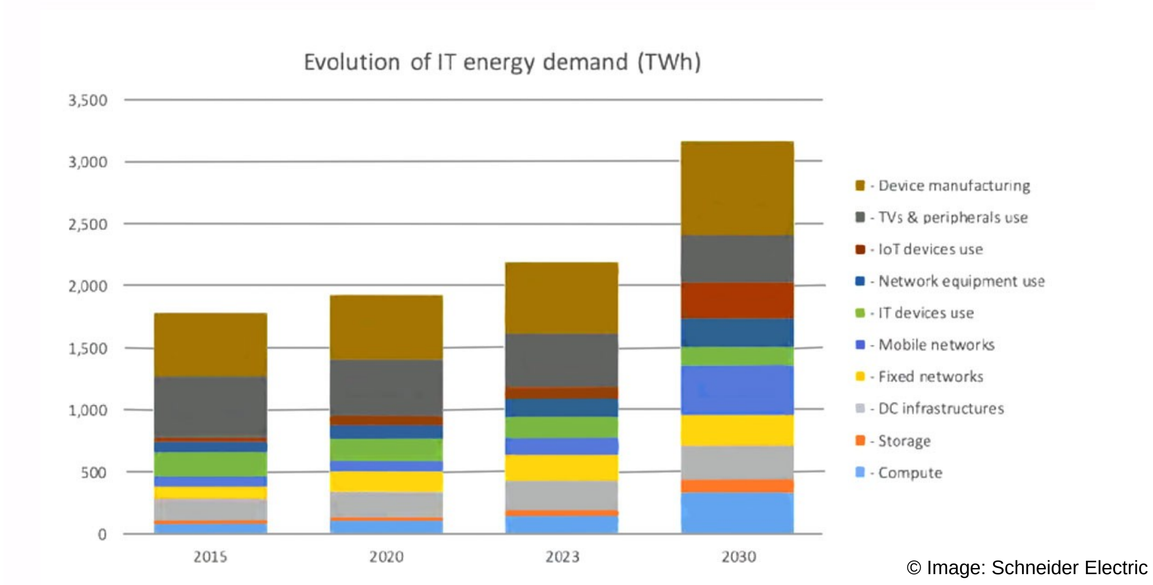
# A FLEXIBLE TOOL FOR ENERGY MEASUREMENT OF PROGRAMS

*Roblex NANA TCHAKOUTE*,
Centre de recherche en informatique (CRI), Mines Paris - PSL
**PhD advisors :** Claude TADONKI (CRI), Petr DOKLADAL (CMM), Youssef MESRI (CEMEF)

## CONTEXT




© Image: Schneider Electric

## MOTIVATION

- **Training AI** : It is estimated that energy consumption due to training GPT-3 is 1287 Mwh, for 552 $tCO_2e$ (tons of $CO_2$ equivalent emissions)
*equivalent to driving 112 gasoline powered cars for a year*

**- Inference AI** : BLOOM, consumed 914 kWh of electricity and emitted 360 kg for 18 days where it handled 230,768 requests (roughly 1.56 $gCO_2e$ per request)
*350 $kgCO_2$ = 1/3 Paris-New-York return*

**- Embedded devices :** The AGX Orin is currently the most powerful board from Nvidia Jetson, with up to 275 TOPS and 60W TDP.
*On full TDP, 60Wh battery will have 1h of life time.*

## BAKGROUND ON ENERGY MEASUREMENT

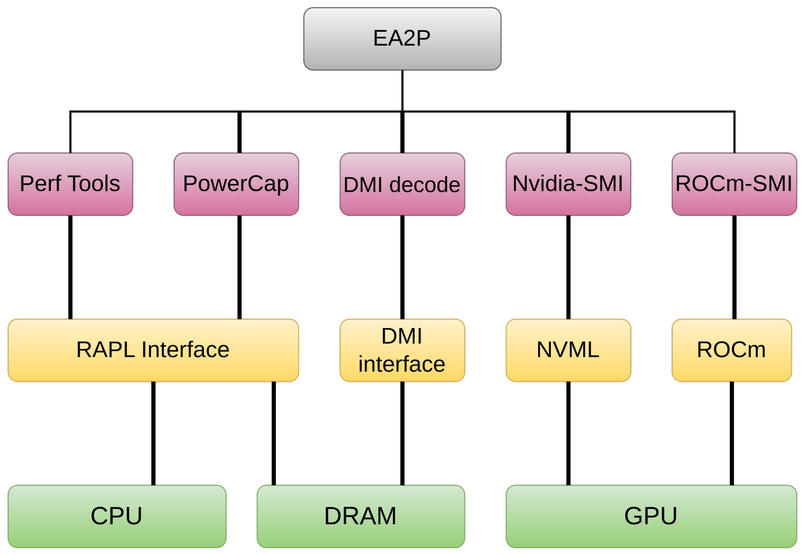
There are hardware sensors that constantly get either the power or the energy of the device (or specific parts) while running and the measurements are stored into specific registers. (they are recent, otherwise we would go with rough and global estimations). They are essential to get power/energy informations

- Intel provided RAPL as embedded energy sensors for CPU grouped in power domains
- AMD provided similar ones for their CPU
- Nvidia GPU have Nvidia-SMI
- AMD GPU have ROCm-SMI
- And so on...

## STATE OF THE ART : ENERGY TOOLS

| Support | CC | EIT | CT | Eco2AI | TB | Pyjoule | Perf | Likwid | PAPI | PG | PT | EA2P(our) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **GPU support** | | | | | | | | | | | | |
| Nvidia GPU | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | ✓ |
| AMD GPU | | | | | | | | | | | | ✓ |
| Intel GPU | | | | | | | | | | | | |
| **CPU and RAM supports** | | | | | | | | | | | | |
| Intel CPU | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| AMD CPU | | | | ✓ | | | ✓ | | | | ✓ | ✓ |
| RAM | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | | ✓ |
| **OS support** | | | | | | | | | | | | |
| Linux | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Windows | ✓ | | | | | | | | | ✓ | | ✓ |
| Mac OS | ✓ | ✓ | | | ✓ | | | | | ✓ | | |
| **Others important characteristics** | | | | | | | | | | | | |
| Documentation | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Configurable | ✓ | ✓ | | | | | | | | | ✓ | ✓ |
| code API | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | | | ✓ |

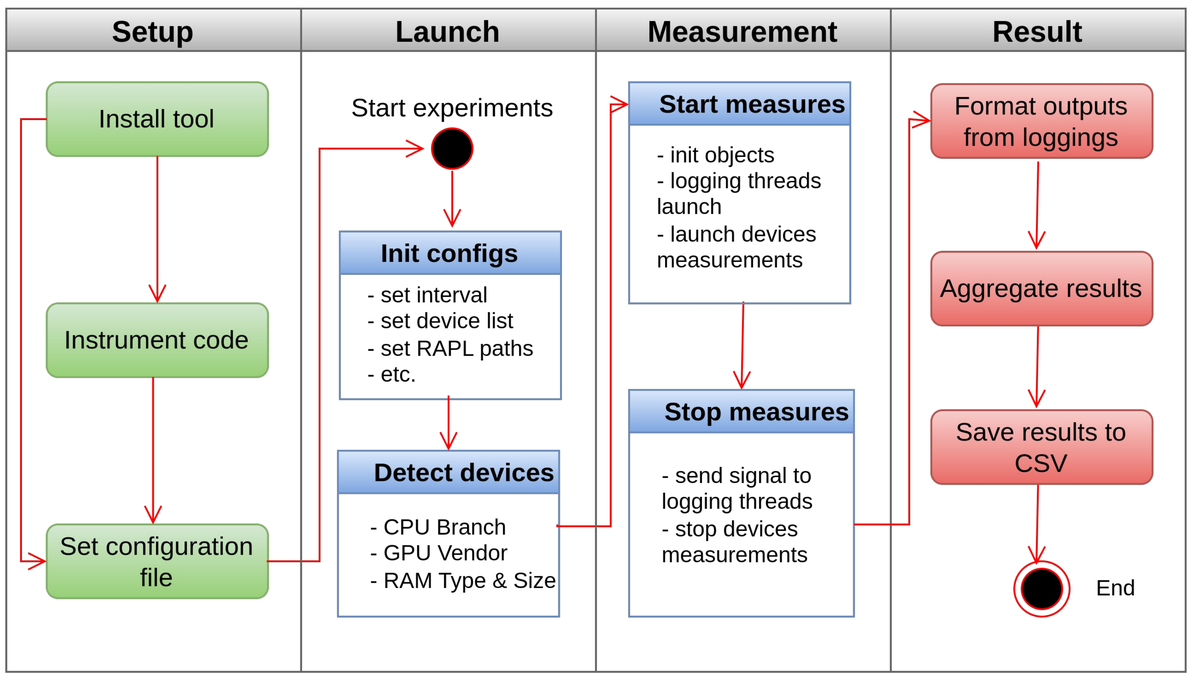
## SOFTWARE ARCHITECTURE OF OUR TOOL




## KEYS CHARACTERISTICS OF A GOOD TOOL

- **Programmability** : Should allows developers to focus on specific parts of the codebase to optimize energy efficiency and performance (instrumentation and APIs)
**- Flexibility :** To measure specific parts of the computer, allowing configurations, auto target hardware detection, and porting to other architectures.
**- Standalone :** Easy to install, few dependences on others library and tools, minimum privileged rights for access
**- Portability :** Compatibility across device generations, even within the same manufacturer (facilitate maintenance)
**- Accuracy :** The tool does indeed measure the desired behavior and should be consistent across workloads

## FUNCTIONAL WORKFLOW




## EXPERIMENTAL VALIDATION

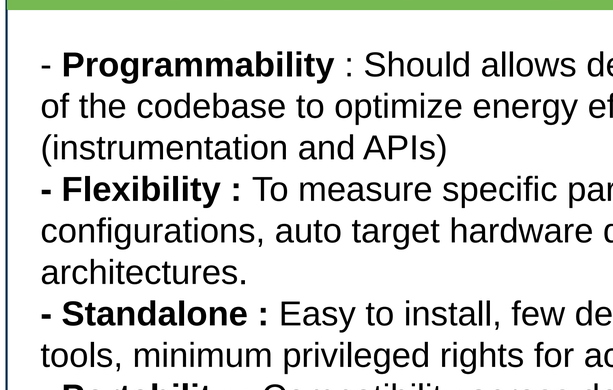



Fig. (a) : Validation of sampling frequency variation

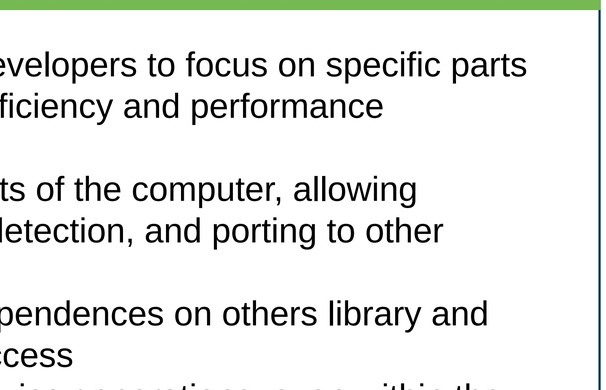



Fig. (b) : Example for multi-chip and multi-threaded application

## ACCESS LINKS

**- GitHub** : https://github.com/HPC-CRI/EA2P
**- Documentation** : https://hpc-cri.github.io/EA2P/