Computing Architecture for Large Language Models (LLMs) and Large Multimodal Models (LMMs)

Bor-Sung Liang

MediaTek
bs.liang@mediatek.com

Department of Computer Science and Information Engineering, EECS
National Taiwan University
bsliang@ntu.edu.tw

Contents

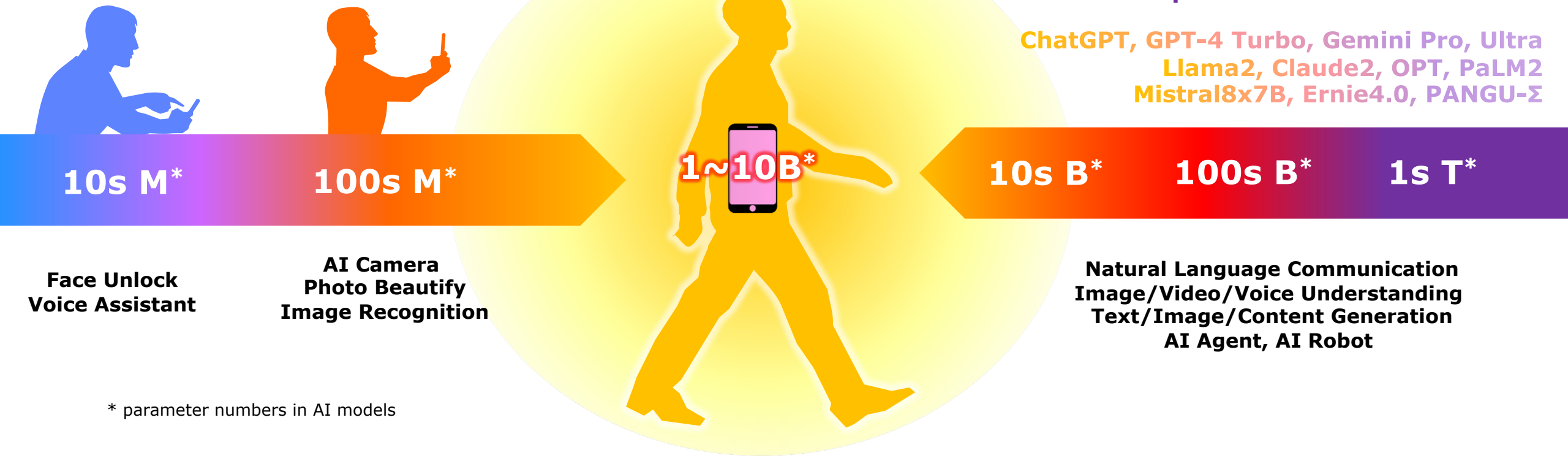
Motivation

AI Inference on Mobile Processors
for
Large Language Models (LLMs)
Large Multimodal Models (LMMs)

On-Device
AI Model Parameter Numbers
Enlarged from 10s Million

Large Language Model
AI Model Parameter Numbers
Up to 100s Billion ~ 1s Trillion

ChatGPT, GPT-4 Turbo, Gemini Pro, Ultra
Llama2, Claude2, OPT, PaLM2
Mistral8x7B, Ernie4.0, PANGU-Σ



Face Unlock
Voice Assistant

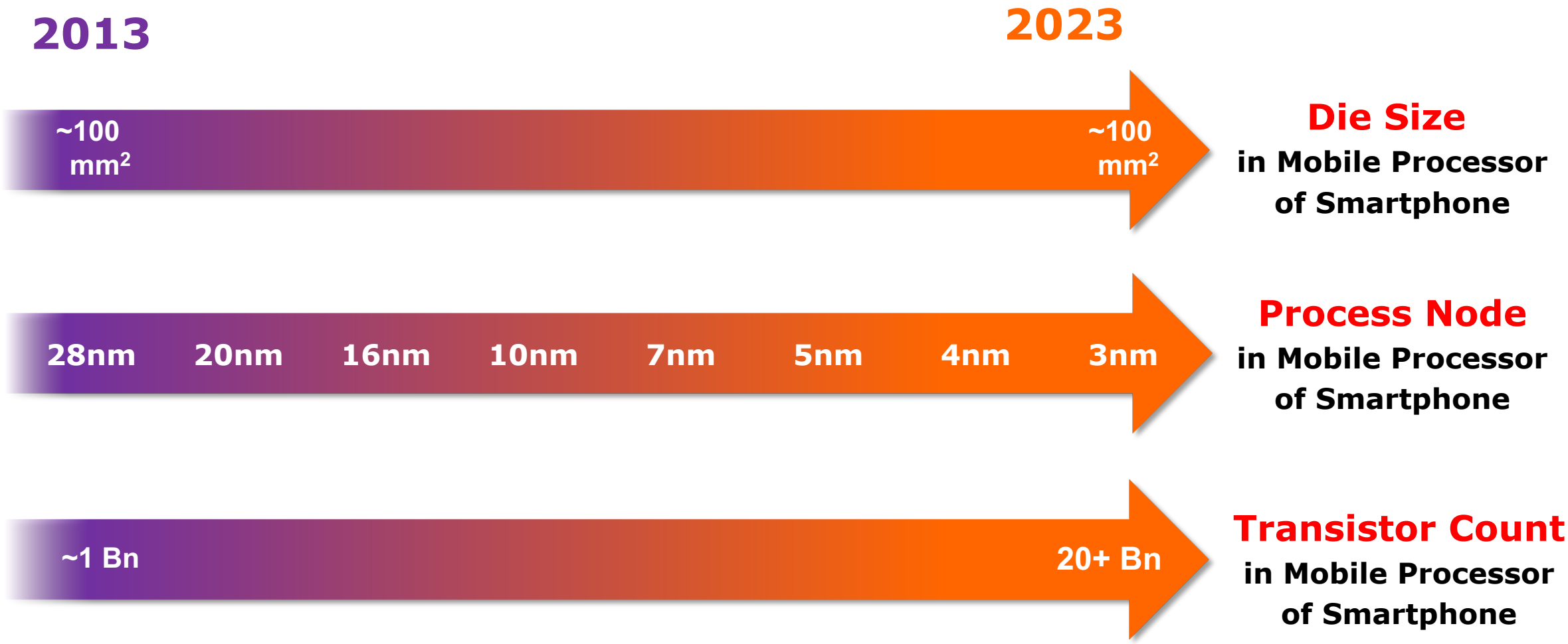
AI Camera
Photo Beautify
Image Recognition

Natural Language Communication
Image/Video/Voice Understanding
Text/Image/Content Generation
AI Agent, AI Robot

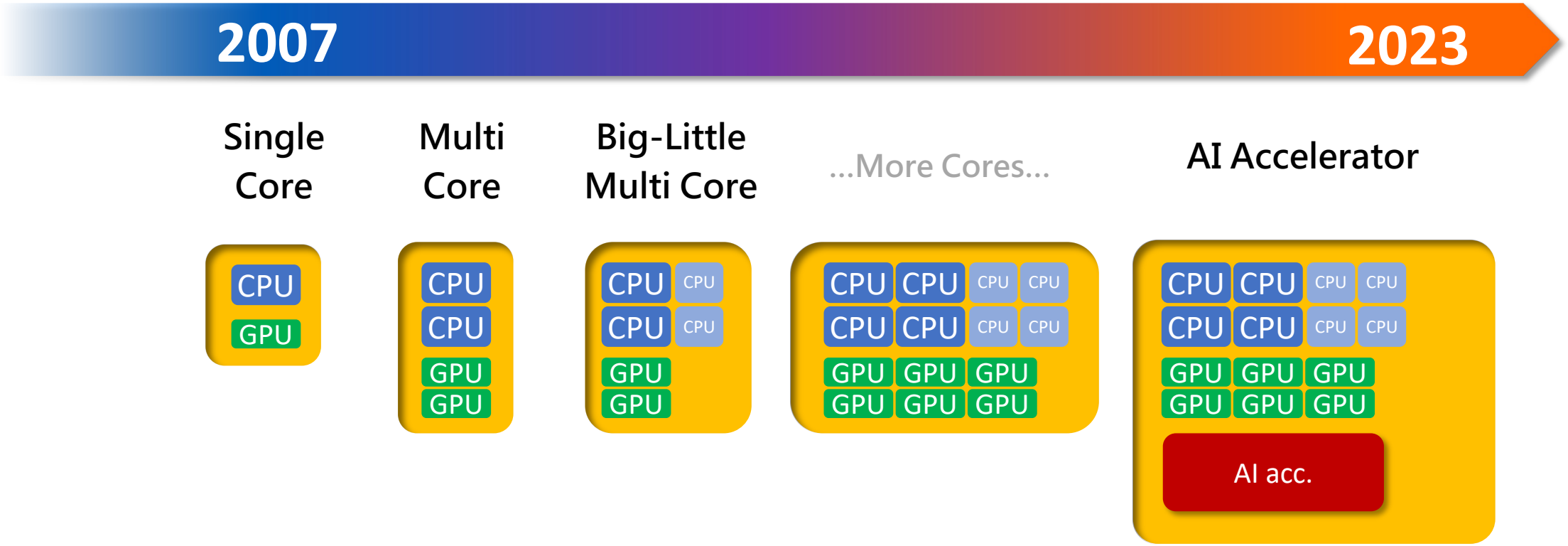
* parameter numbers in AI models

Trends of Mobile Processors and AI Models

Trends of Mobile Processor in a Decade



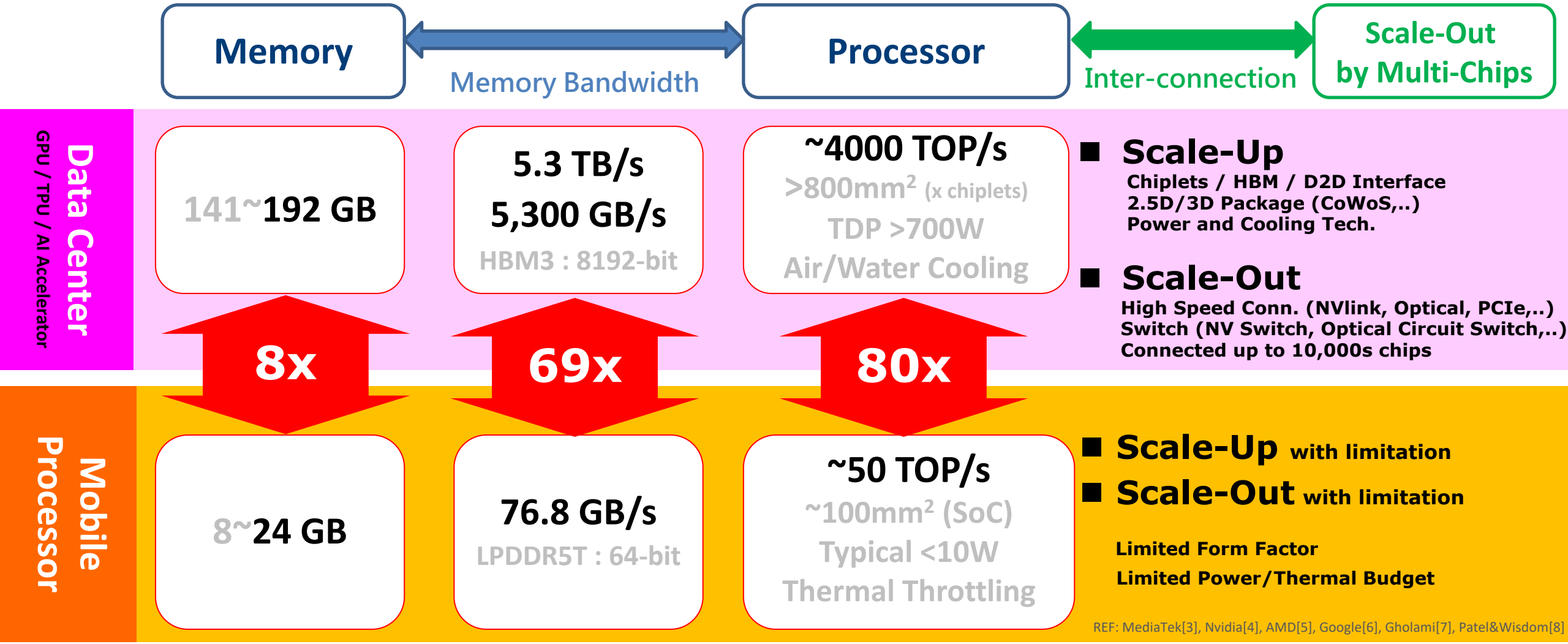
Trends of Computing Architecture in Mobile Processor



- CPU Center Processor Unit
- GPU Graphics Processor Unit
- AI acc. AI Accelerator

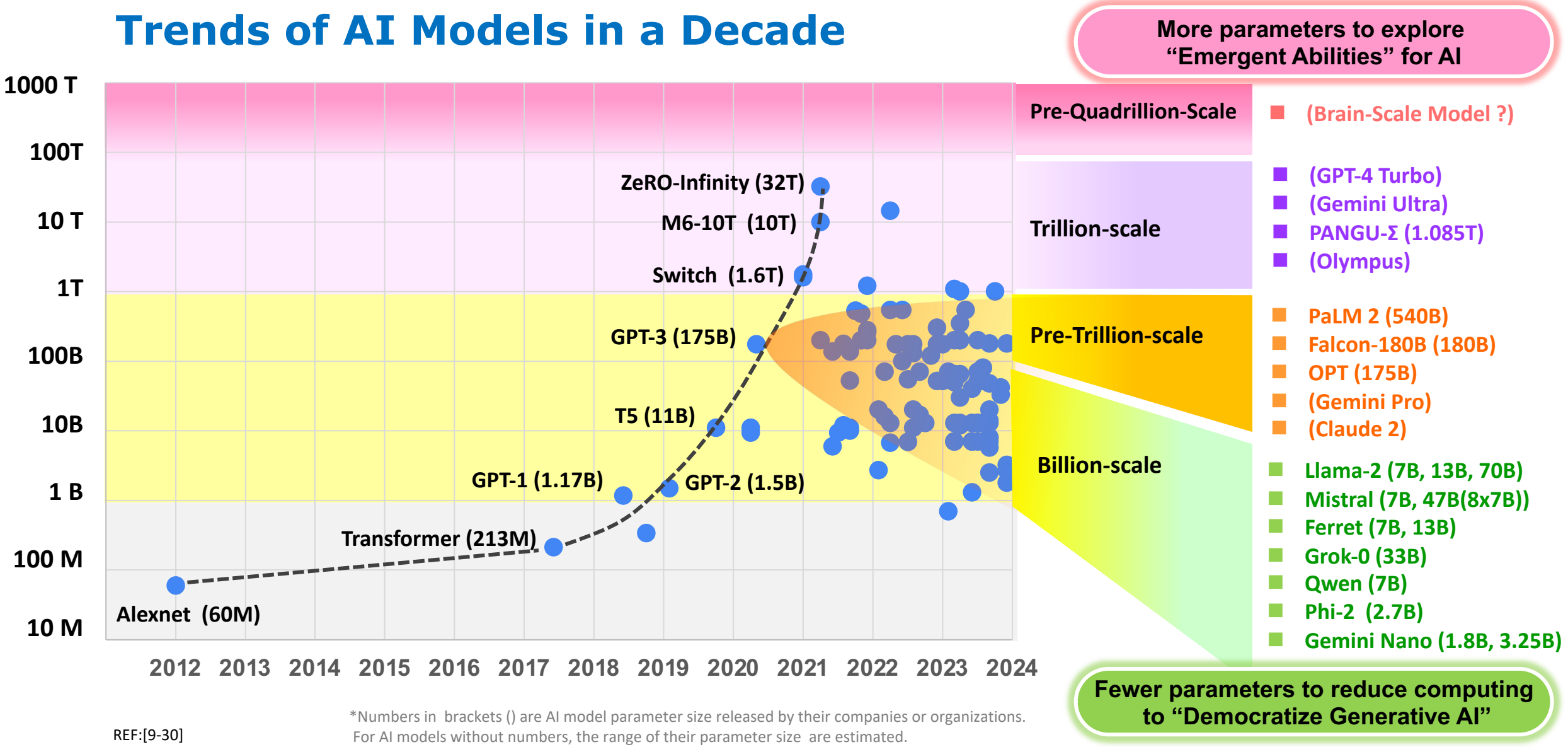
*The number of CPU, GPU, and AI accelerator cores is for reference. Each application processor will have different architecture and number depending on the actual situation.

More Limitations on Mobile Processor for AI Computing



REF: MediaTek[3], Nvidia[4], AMD[5], Google[6], Gholami[7], Patel&Wisdom[8]

Trends of AI Models in a Decade

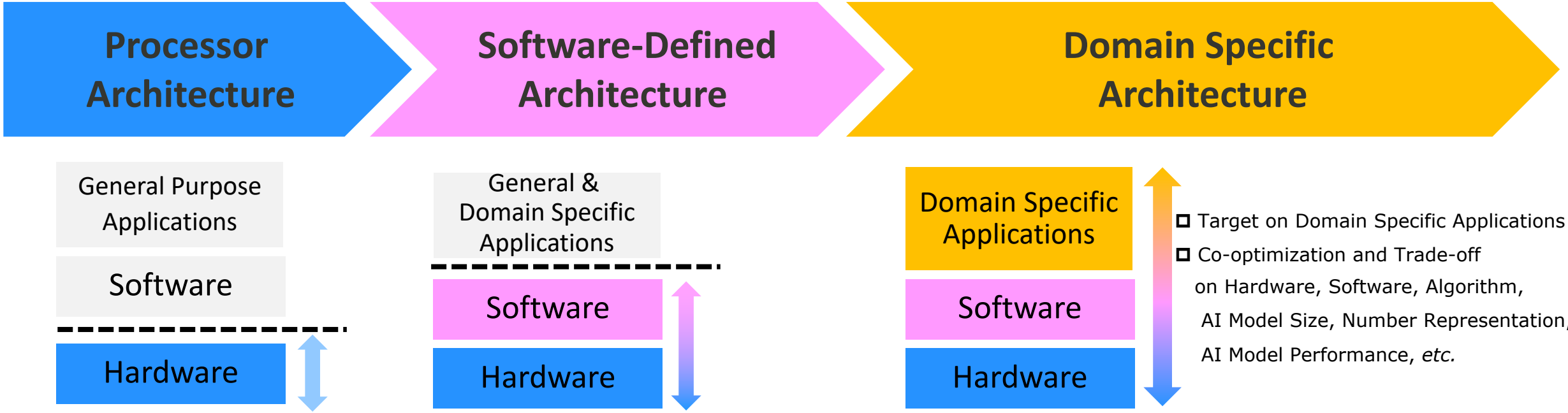


REF:[9-30]

*Numbers in brackets () are AI model parameter size released by their companies or organizations. For AI models without numbers, the range of their parameter size are estimated.

Domain Specific Architecture Design for AI Inference

Trends for Architecture Design : Domain Specific Architecture



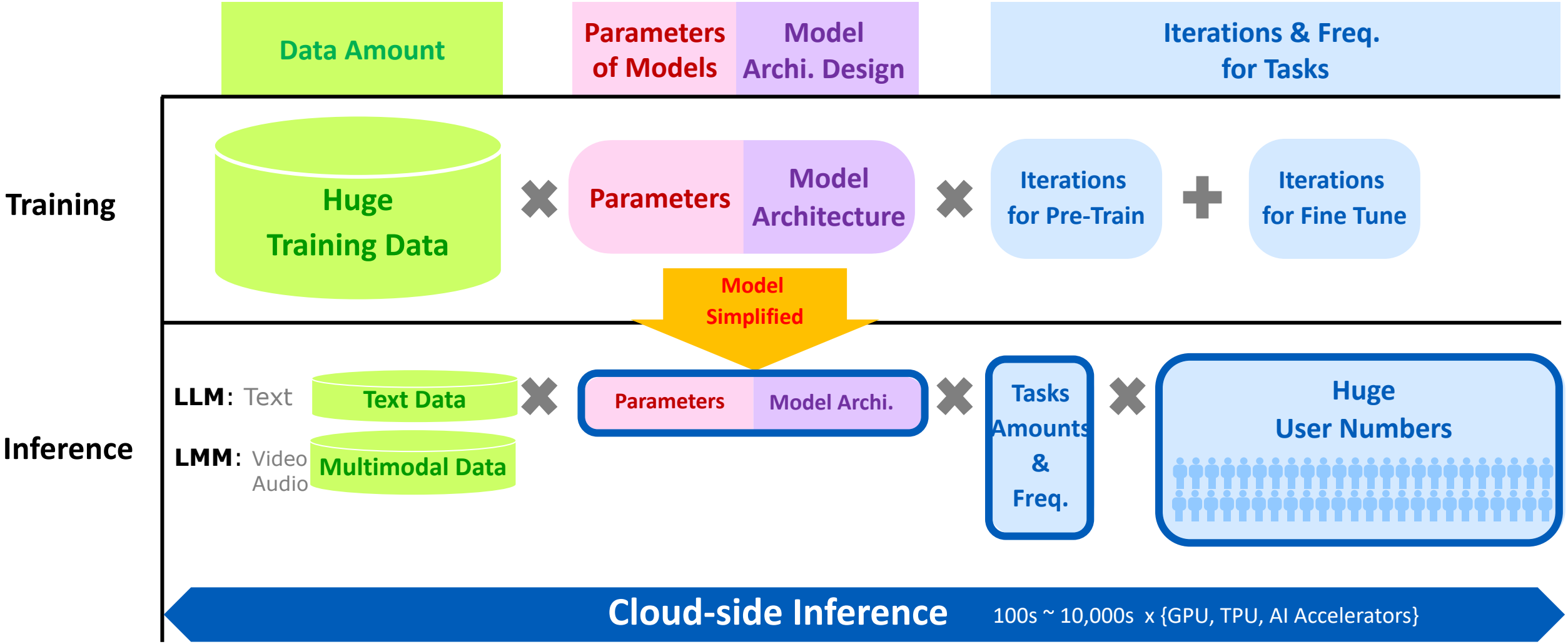
- **Hardware Design**
- **High Speed Circuit Design**
- **Semi. Process/System**
  (More Moore, More-than-Moore, and Beyond Moore)

- **Hardware-Software Co-design**
- **Reconfigurable/Programmable**
- **Customized Instruction/Data Flow**

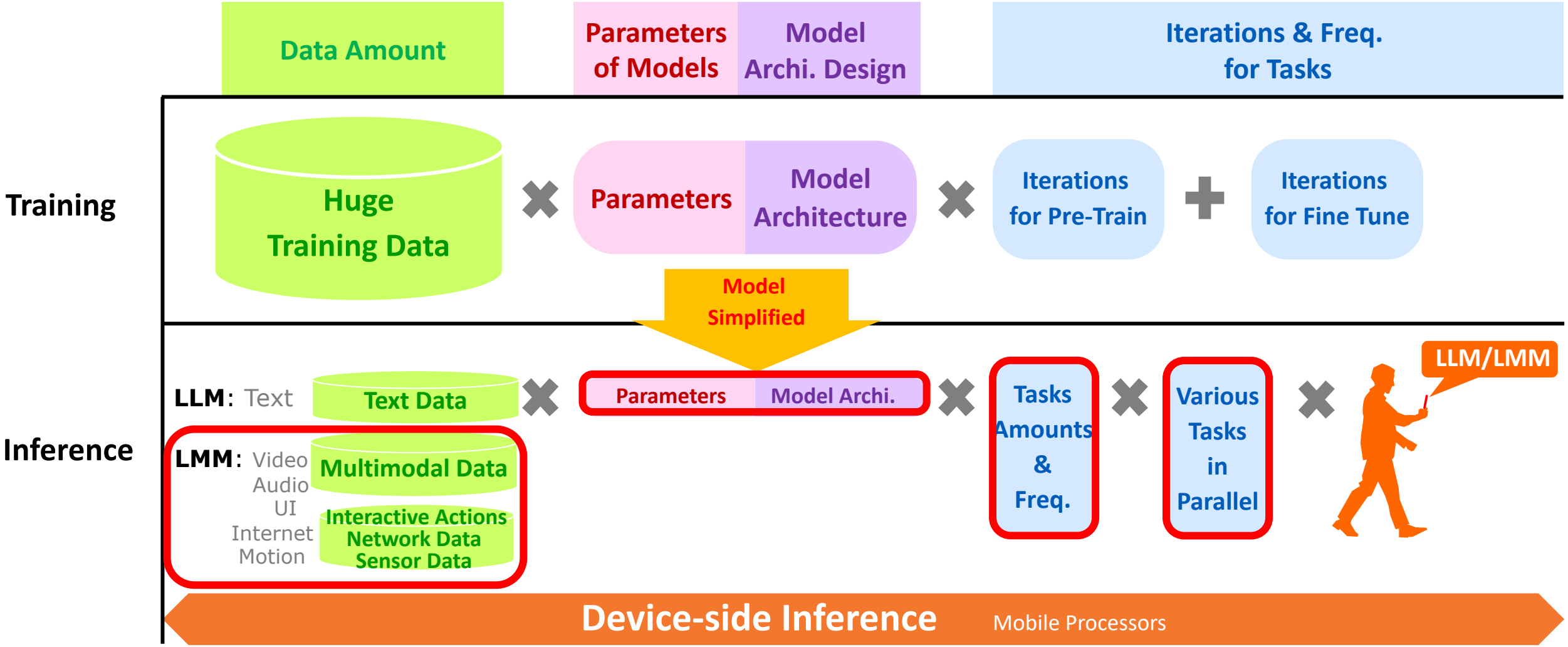
- **Hardware-Software-Domain Co-design**
- **Co-optimized with AI Model**
- **Trade-off to Fulfill Huge AI Computing Demand**

REF: Hennessy & Patterson [1], IRDS [2]

AI Computing for Cloud-side Inference



AI Computing for Device-side Inference



Techniques to Reduce AI Model Size

- **To Reduce Memory Footprint in Storage and Computing Cost**
  - **Number Representation**
    - Quantization / Reducing the precision
    - 32-bit (FP32) → 16-bit (FP16, BF16) → 8-bit (INT8, FP8) → 4-bit (INT4) …
  - **Use Smaller LLM/LMM by Knowledge Distillation (KD)**
    - Utilize a full-size model (teacher) to train a smaller model (student)
  - **Sparsity / Pruning**
    - Increase zero weights in model parameters

REF: Dally[31], Dean&Vahdat[32] , Hinton *et al* [33], Naffziger[34]

* not an exhaustive list

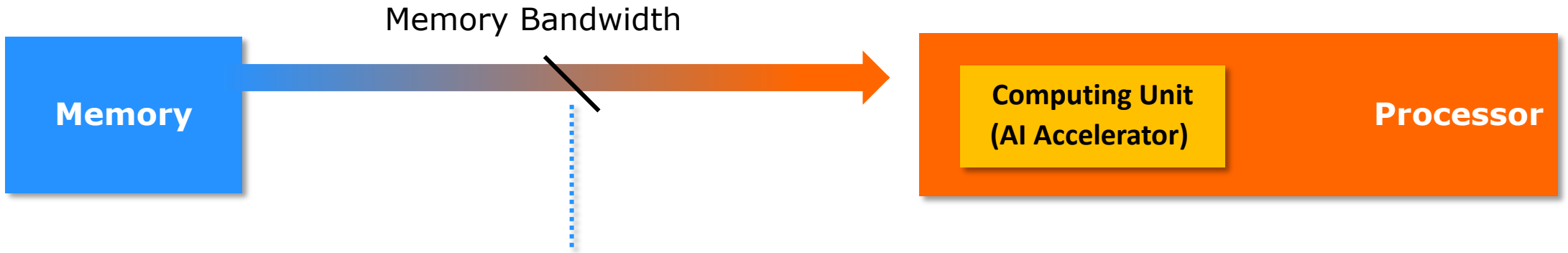
Model Size for different Parameter and Number Representation

Model Parameters	FP32 (typically for Training)	FP16, BF16 (typically for Training)	INT8 (typically for Inference)	INT4 (typically for Inference)
	32bit (4Byte)	**16bit** (2Byte)	**8bit** (1Byte)	**4bit** (0.5Byte)
70 Bn	**280 GB**	**140 GB**	**70 GB**	**35 GB**
13 Bn	**52 GB**	**26 GB**	**13 GB**	**7.5 GB**
7 Bn	**28 GB**	**14 GB**	**7 GB**	**3.5 GB**
3.25 Bn	**13 GB**	**6.5 GB**	**3.25 GB**	**1.625 GB**
1.8 Bn	**7.2 GB**	**3.6 GB**	**1.8 GB**	**0.9 GB**

Single digital Billion-Scale Parameter LLM

DRAM Size : >128GB | 128~24GB | 24~8GB | 8~4GB | 4~2GB | <2GB

Bandwidth Utilization Ratio



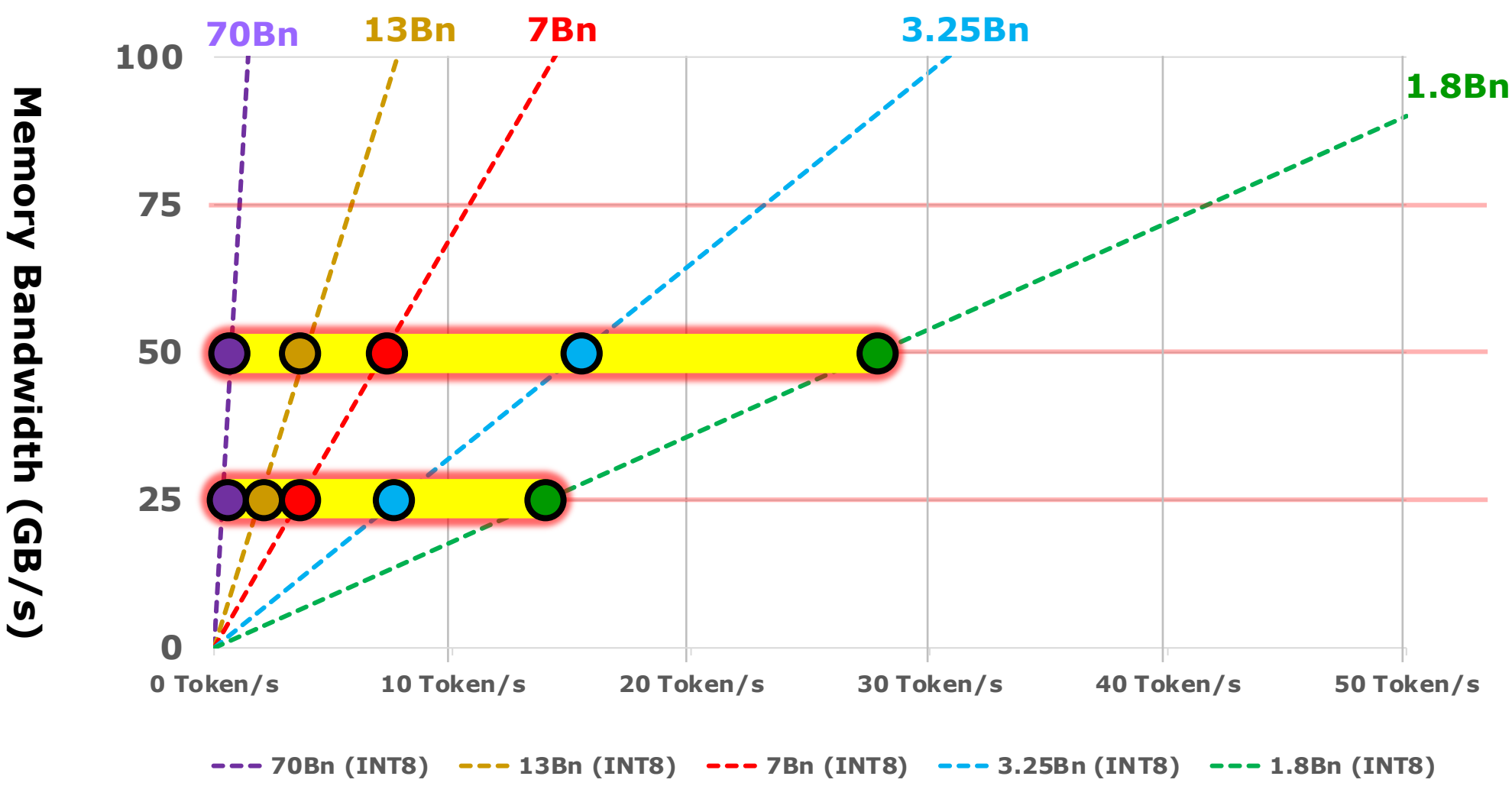
MBU

Model Bandwidth Utilization

$$\frac{\text{Real memory bandwidth for LLM}}{\text{Peak memory bandwidth}}$$

REF: Agarwal, *et al* [35]

Token Speed under Memory Bandwidth Limit (LLM @ INT8)



Memory:
LPDDR5T-9600
(9.6Gbps, 64bit)
76.8 GB/s

100%
Model Bandwidth Utilization

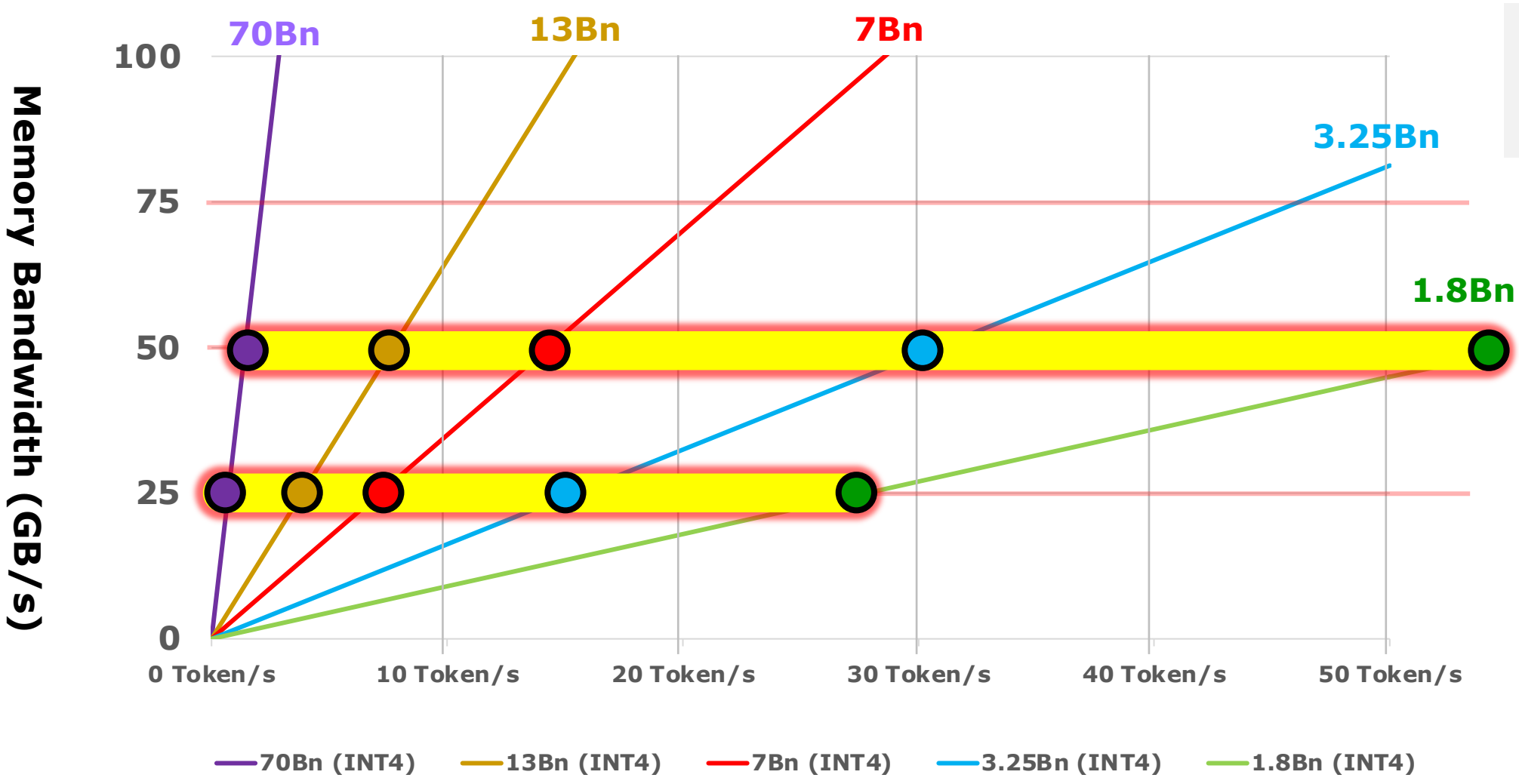
65%
Model Bandwidth Utilization

33%
Model Bandwidth Utilization

* Trivial LLM porting without optimize model for bandwidth

- Only consider Bandwidth of "Memory to Processor"

Token Speed under Memory Bandwidth Limit (LLM @ INT4)



Memory:
LPDDR5T-9600
(9.6Gbps, 64bit)
76.8 GB/s

100%
Model Bandwidth Utilization

65%
Model Bandwidth Utilization

33%
Model Bandwidth Utilization

* Trivial LLM porting without optimize model for bandwidth

- Only consider Bandwidth of "Memory to Processor"

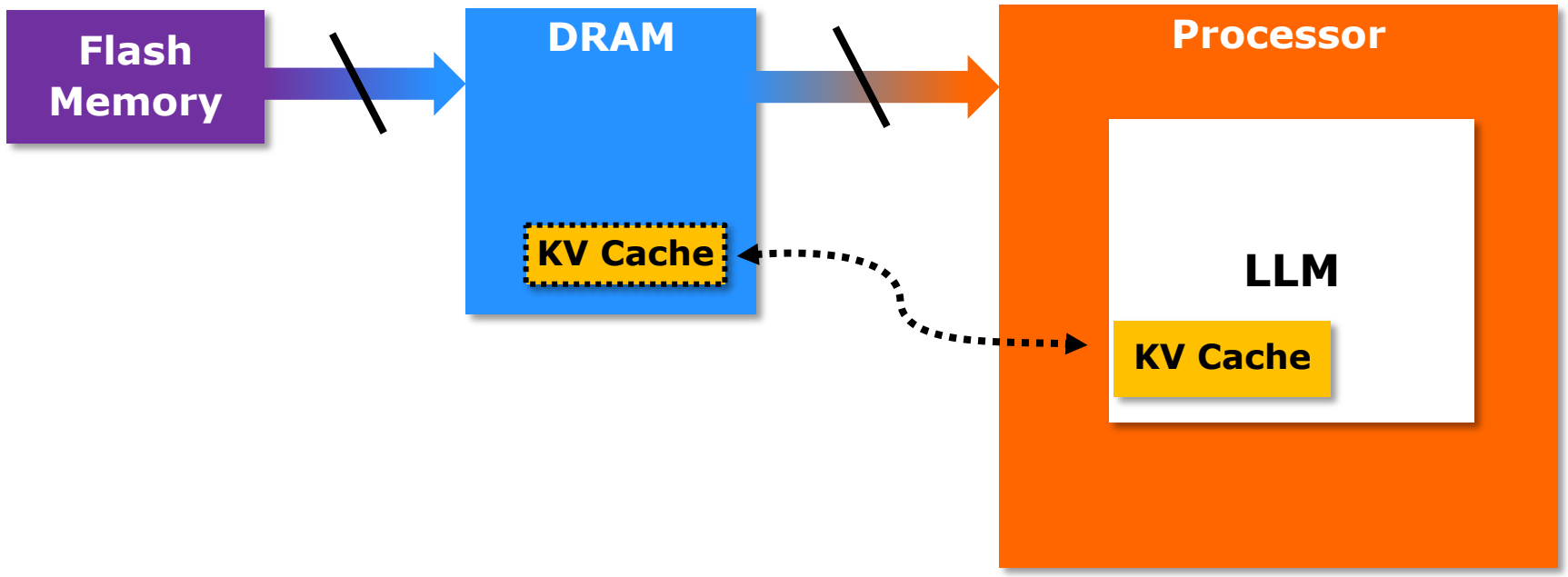
Techniques to Improve LLM/LMM Performance

- **KV Cache (Key/Value cache)**
- **Retrieval Augmented Generation (RAG)**
- **Sparsity Mixture of Experts (SMoE)**
- **Speculative Execution**
- **Speculative Execution with Cloud-Device Collaboration**

* not an exhaustive list

KV Cache (Key/Value Cache)

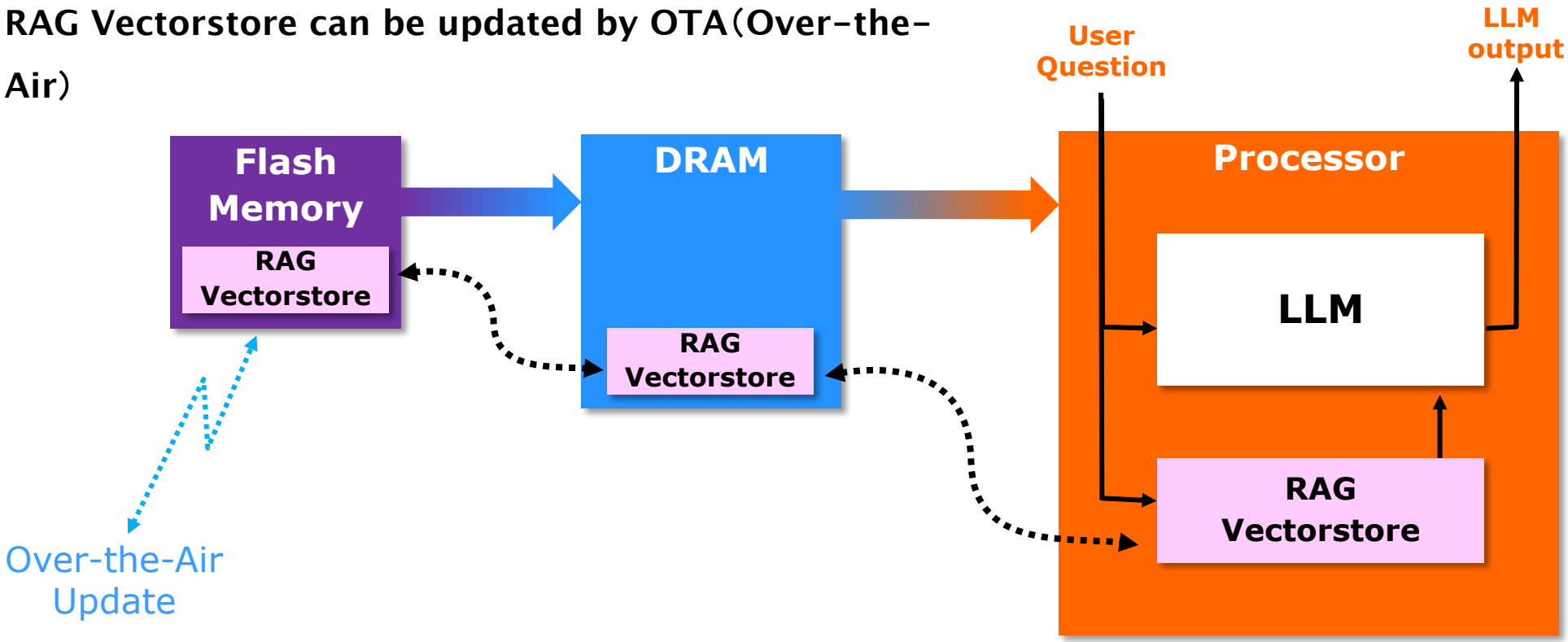
- **Catch previous Keys and Values to save recalculation**



REF: Agarwal, *et al* [35], Chen [36]

Retrieval Augmented Generation (RAG)

- Incorporating knowledge from external databases
- Enhances the accuracy and credibility of the models
- RAG Vectorstore can be updated by OTA(Over-the-Air)



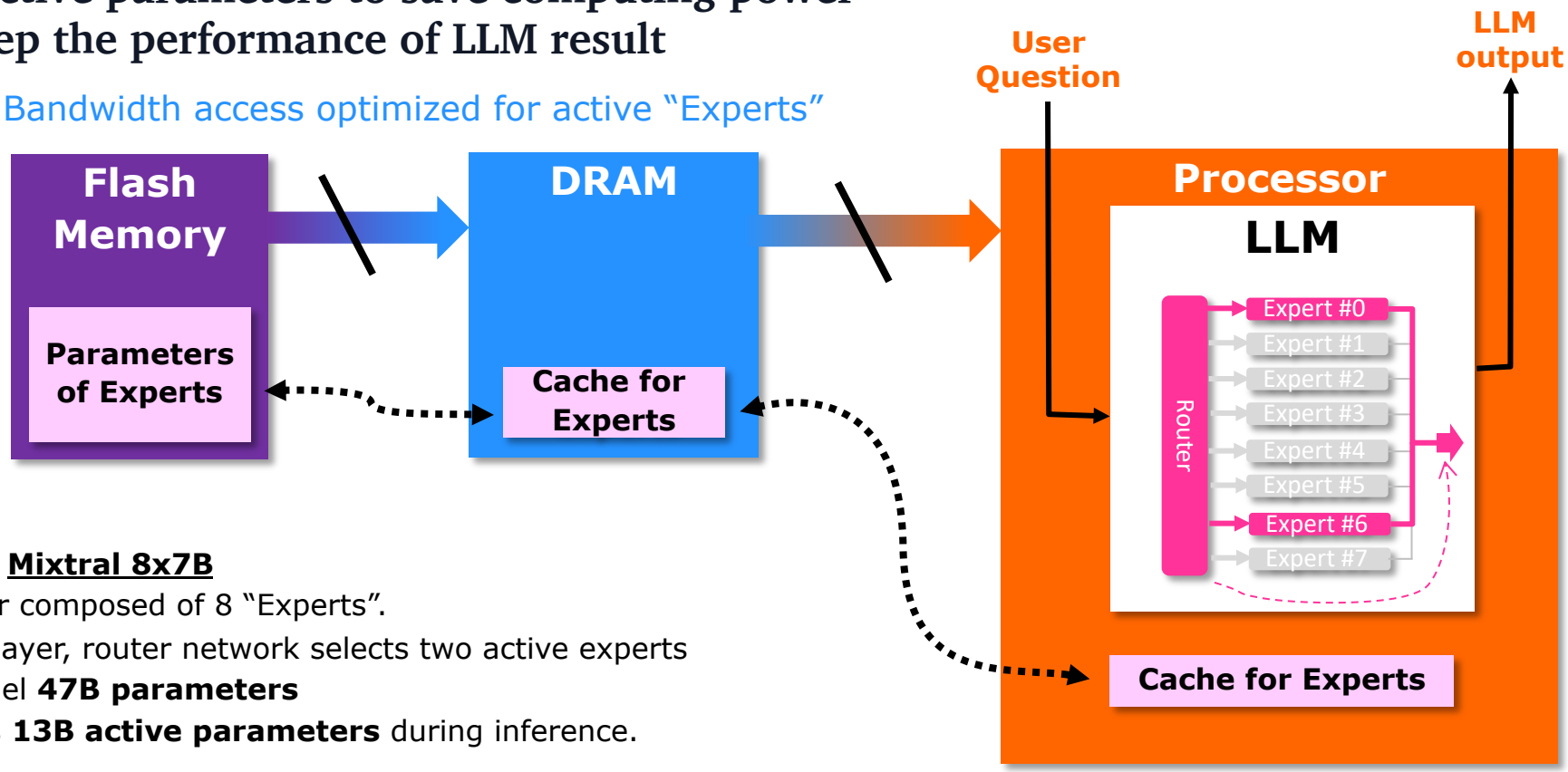
REF: Lewis , *et al.*[37]

Sparse Mixture of Experts (SMoE)

- Layers have a certain number of “Experts”
- Router to determine which “Experts” are active
- Fewer active parameters to save computing power but keep the performance of LLM result

REF: Gao , *et al.*[31]

* Bandwidth access optimized for active “Experts”



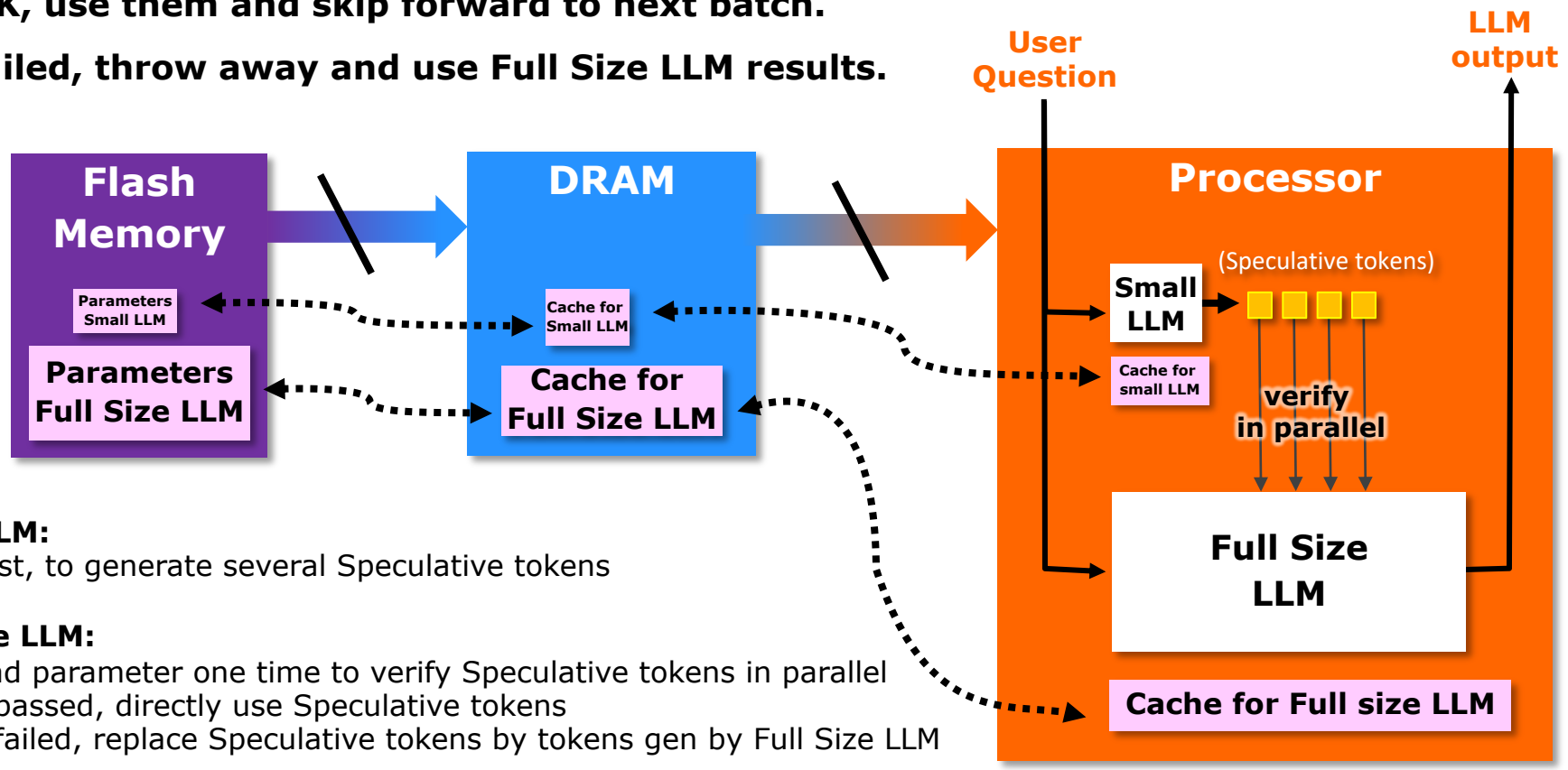
For example: **Mixtral 8x7B**

- each layer composed of 8 “Experts”.
- For each layer, router network selects two active experts
- Total model **47B parameters**
- Only uses **13B active parameters** during inference.

REF: Eliseev&Mazur[38], Mistral.AI [39]

Speculative Execution

- **Small LLM to generate Speculative tokens**
- **Feed all Speculative tokens to Full Size LLM in a batch to verify**
  - **If OK, use them and skip forward to next batch.**
  - **If failed, throw away and use Full Size LLM results.**

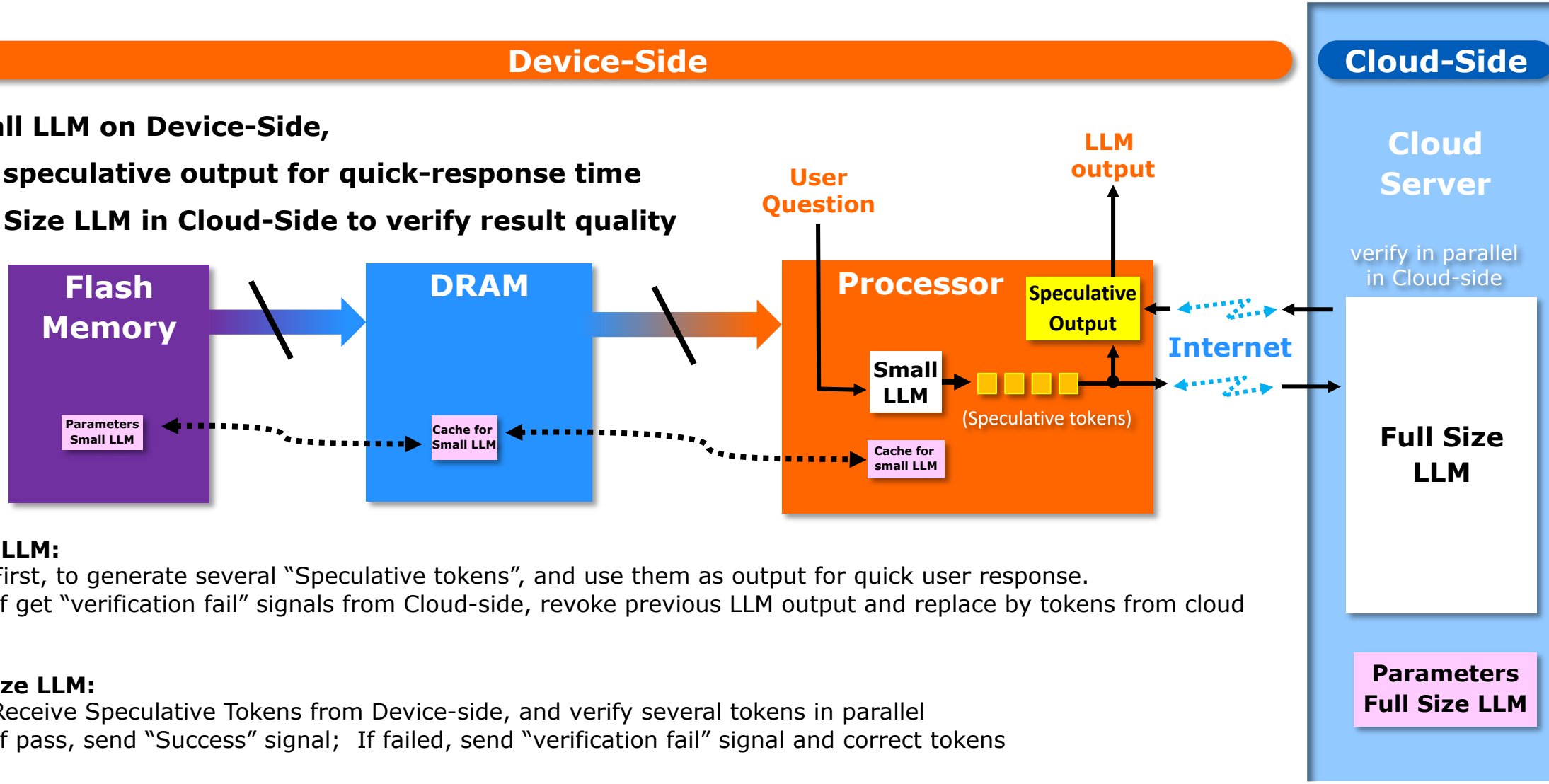


- **Small LLM:**
  - First, to generate several Speculative tokens
- **Full Size LLM:**
  - load parameter one time to verify Speculative tokens in parallel
  - If passed, directly use Speculative tokens
  - If failed, replace Speculative tokens by tokens gen by Full Size LLM

- Output verified result
- If speculative tokens failed, they will be replaced by token generated by Full Size LLM.

REF: Karpathy [40], Stern, *et al* [41], Leviathan, *et al* [42]

Speculative Execution (Cloud-Device Collaboration)



- **Small LLM:**
  - First, to generate several "Speculative tokens", and use them as output for quick user response.
  - If get "verification fail" signals from Cloud-side, revoke previous LLM output and replace by tokens from cloud

- **Full Size LLM:**
  - Receive Speculative Tokens from Device-side, and verify several tokens in parallel
  - If pass, send "Success" signal; If failed, send "verification fail" signal and correct tokens

REF: Karpathy [40], Stern, *et al* [41], Leviathan, *et al* [42]

Token Speed

for LLM & LMM

How Fast We Read

Read word-by-word

Read Skimming looking *only* for the general or main ideas
Read Scanning look *only* for a specific information

Read Normally 250 words/min

(Adults Silent-reading Speed 238 words/min)

The International Solid-State Circuits Conference (ISSCC) is an esteemed annual gathering that serves as a global platform for the presentation of cutting-edge advancements in solid-state circuits and systems-on-a-chip. It is renowned for showcasing the latest research and breakthroughs in integrated circuits, drawing engineers, researchers, and industry professionals from across the globe.

At ISSCC, the technology sessions cover a wide range of topics, including:

- Advanced semiconductor technologies
- Innovative circuit designs and architectures
- Trends in system-on-chip (SoC) integration
- High-speed communication and data processing
- Low-power and energy-efficient circuitry
- Silicon photonics and optoelectronic devices
- Novel sensor technologies
- Emerging memory technologies

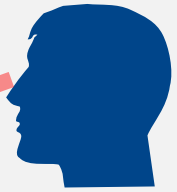
Read Proficiently 1000 words/min

(Skimming speeds of 1000+ words/min)

The International Solid-State Circuits Conference (ISSCC) is an esteemed annual gathering that serves as a global platform for the presentation of cutting-edge advancements in solid-state circuits and systems-on-a-chip. It is renowned for showcasing the latest research and breakthroughs in integrated circuits, drawing engineers, researchers, and industry professionals from across the globe.

At ISSCC, the technology sessions cover a wide range of topics, including:

- Advanced semiconductor technologies
- Innovative circuit designs and architectures
- Trends in system-on-chip (SoC) integration
- High-speed communication and data processing
- Low-power and energy-efficient circuitry
- Silicon photonics and optoelectronic devices
- Novel sensor technologies
- Emerging memory technologies

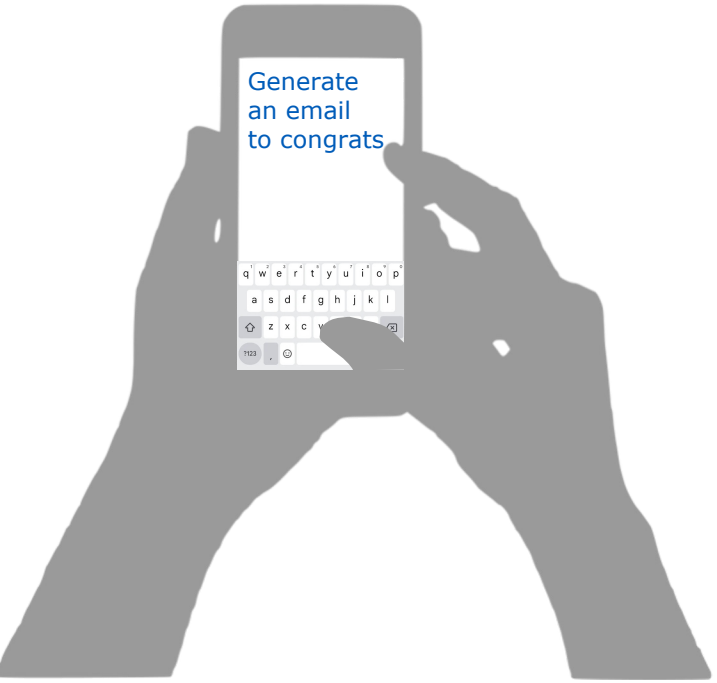


REF: Brysbaert [43]

LLM and LMM in Mobile Devices

LLM (Large Language Model)

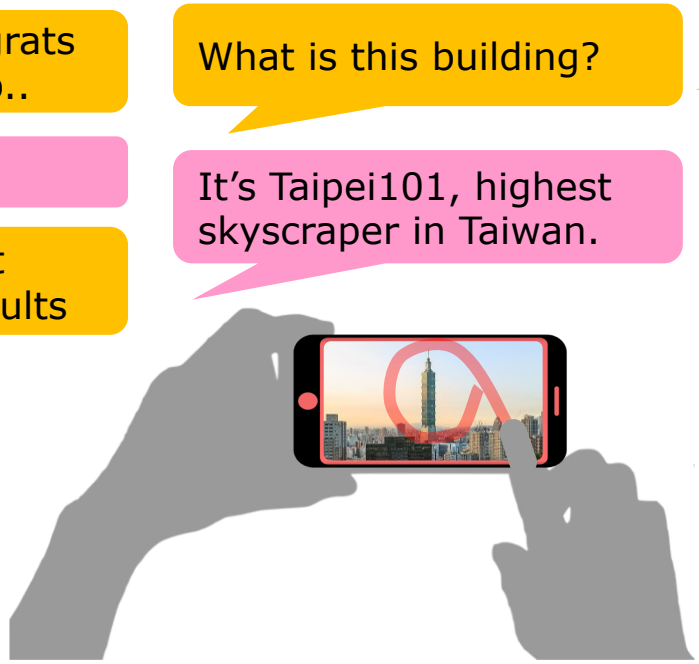
- Typing Text



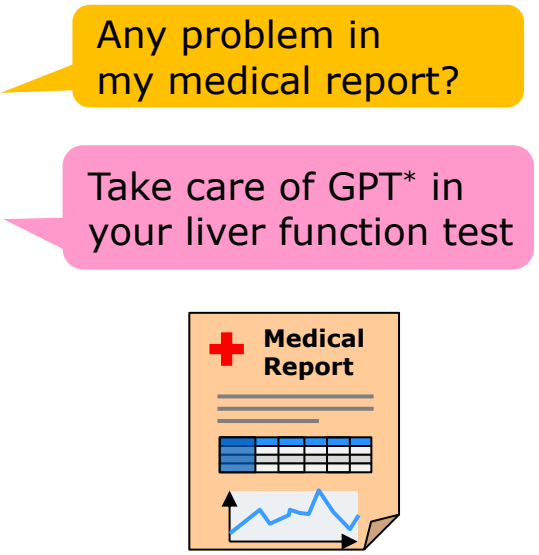
LMM (Large Multimodal Model)

- Voice
- Nature Language Communication

- Camera
- Visual indicator



- Image
- Doc, Table



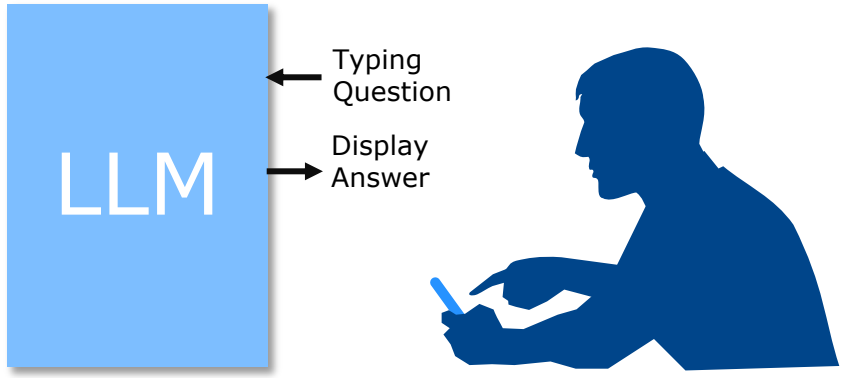
* GPT (glutamyl pyrubic transaminase) in liver function test

Types of Tokens in LLM & LMM

Simple LLM

Single Person

One Simple LLM Task

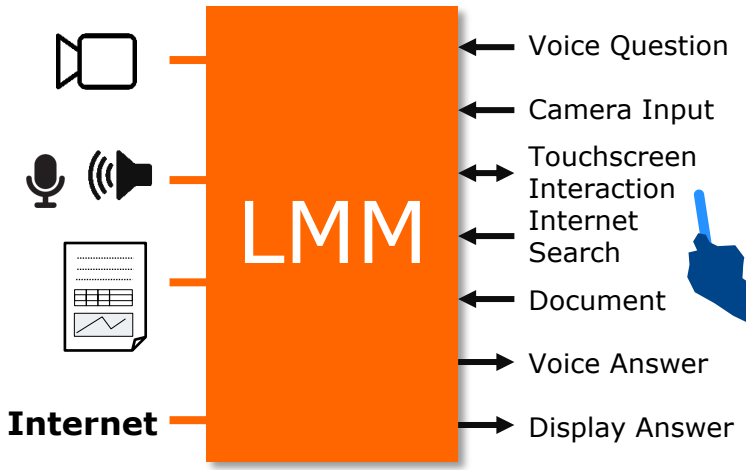


Only Tokens for Text

Read Normally
250 words/min
~8.33 Token/sec

Read Proficiently
1000 words/min
~33.3 Token/sec

Multimodal Interaction

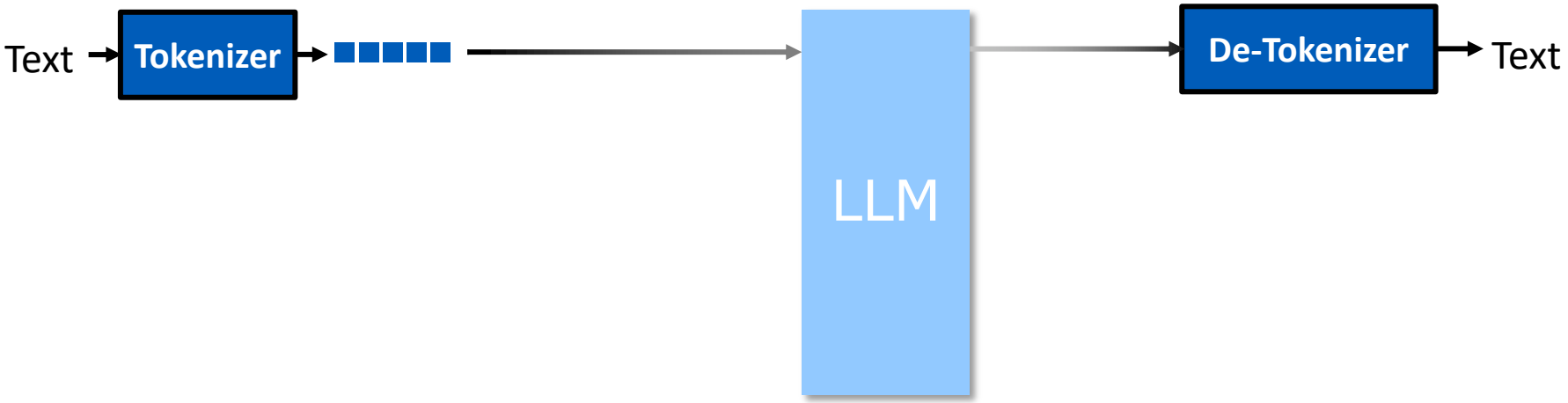


More Various Tokens Concurrently

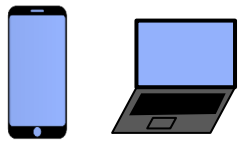
- **Tokens for Text**
- **Tokens for Image/Video**
- **Tokens for Voice /Audio**
- **Tokens for User Interactive**
- **Tokens for Internet Contents**
- **Tokens for Document Contents**

REF: [44-47]

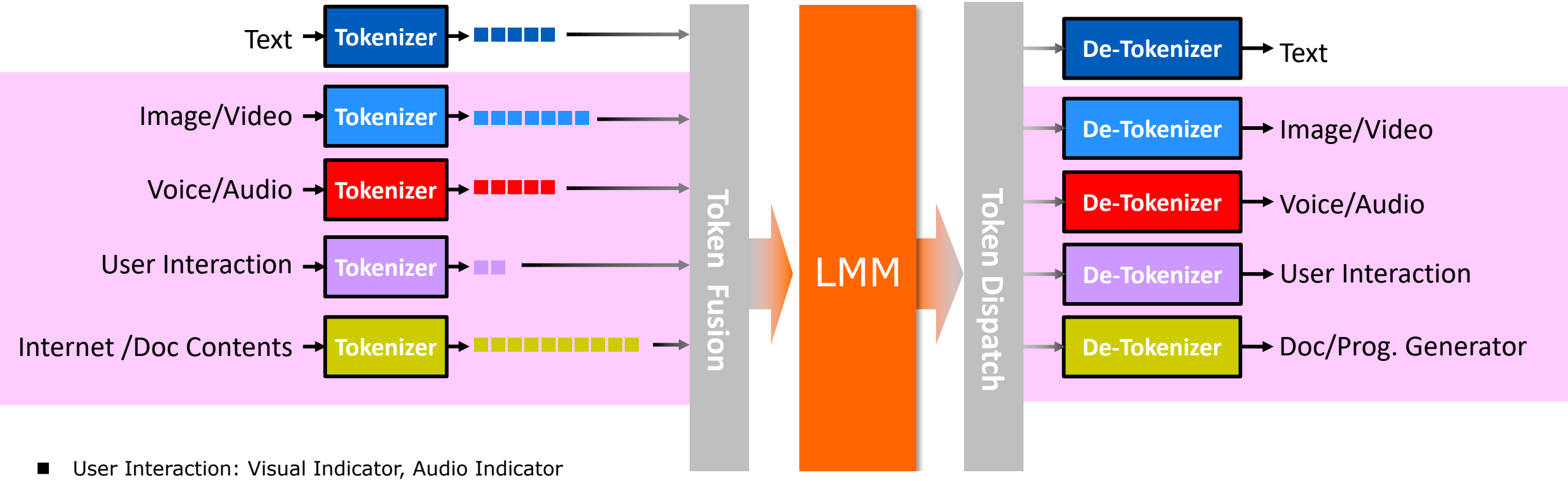
Token Process of LLM



Mobile Devices
w/ text-based LLM

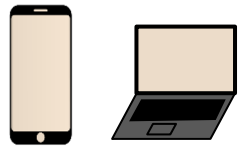


Token Process of LMM for Multimodal Generative AI



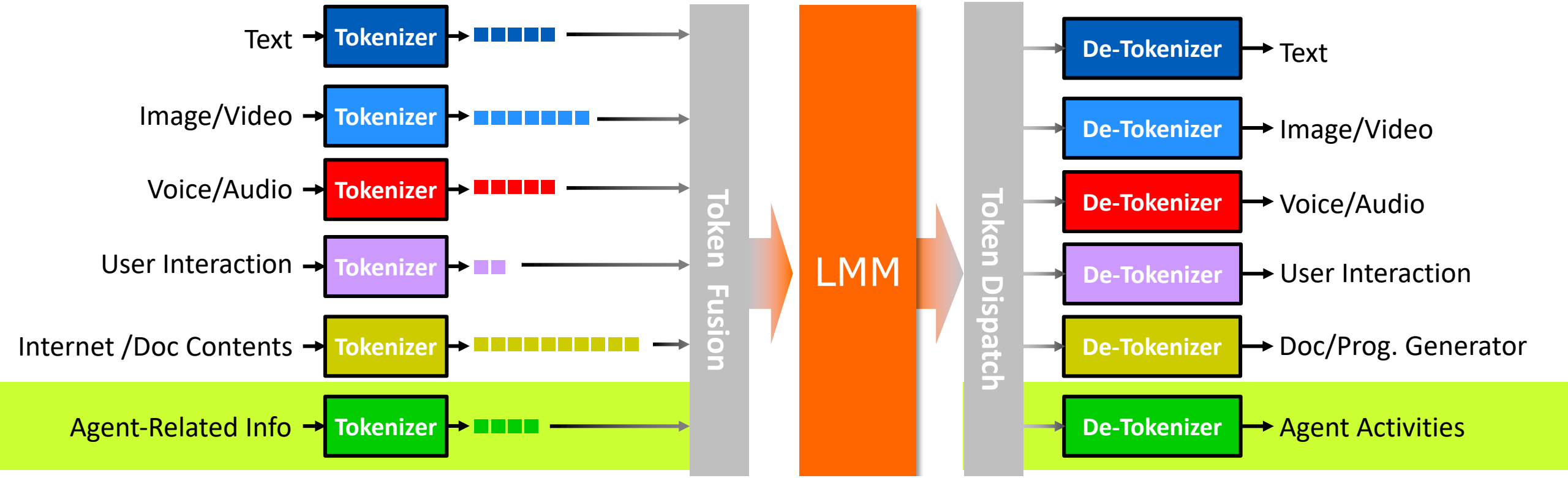
- User Interaction: Visual Indicator, Audio Indicator
- Doc. & Internet Contents - Understanding & Summary
- Application Example : Virtual Assistants, Chatbots, Virtual Tutorr, Translator, Simultaneous Interpretation

Mobile Devices
w/ Multimodal Gen AI



REF: [44-47]

Token Process of LMM for AI Agent

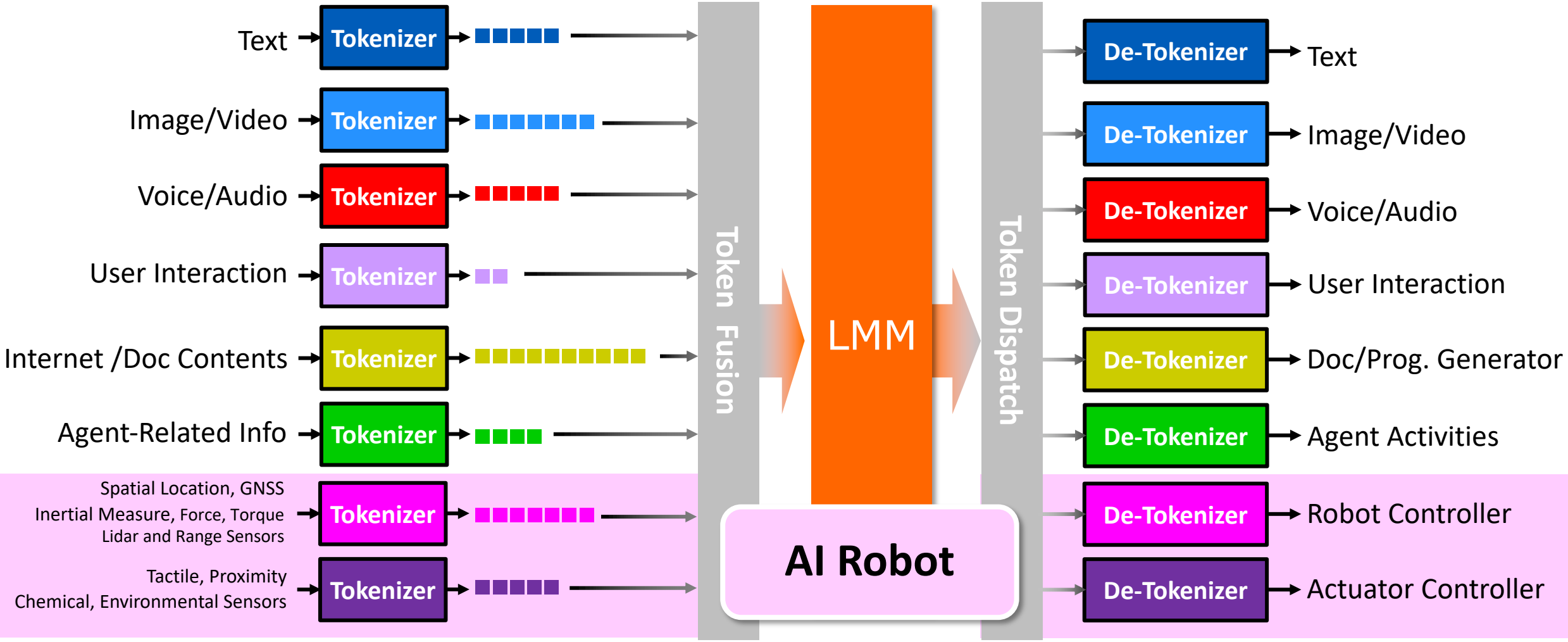


- AI Agent represents the user to operate devices or to interact on Internet
- Issues: Security, Authorization, Trust, Responsibility
  Anti-Spoof, Anti-Theft, Anti-Hallucination

AI Agent

REF: Yan, *et al* [48], Durante, *et al* [49]

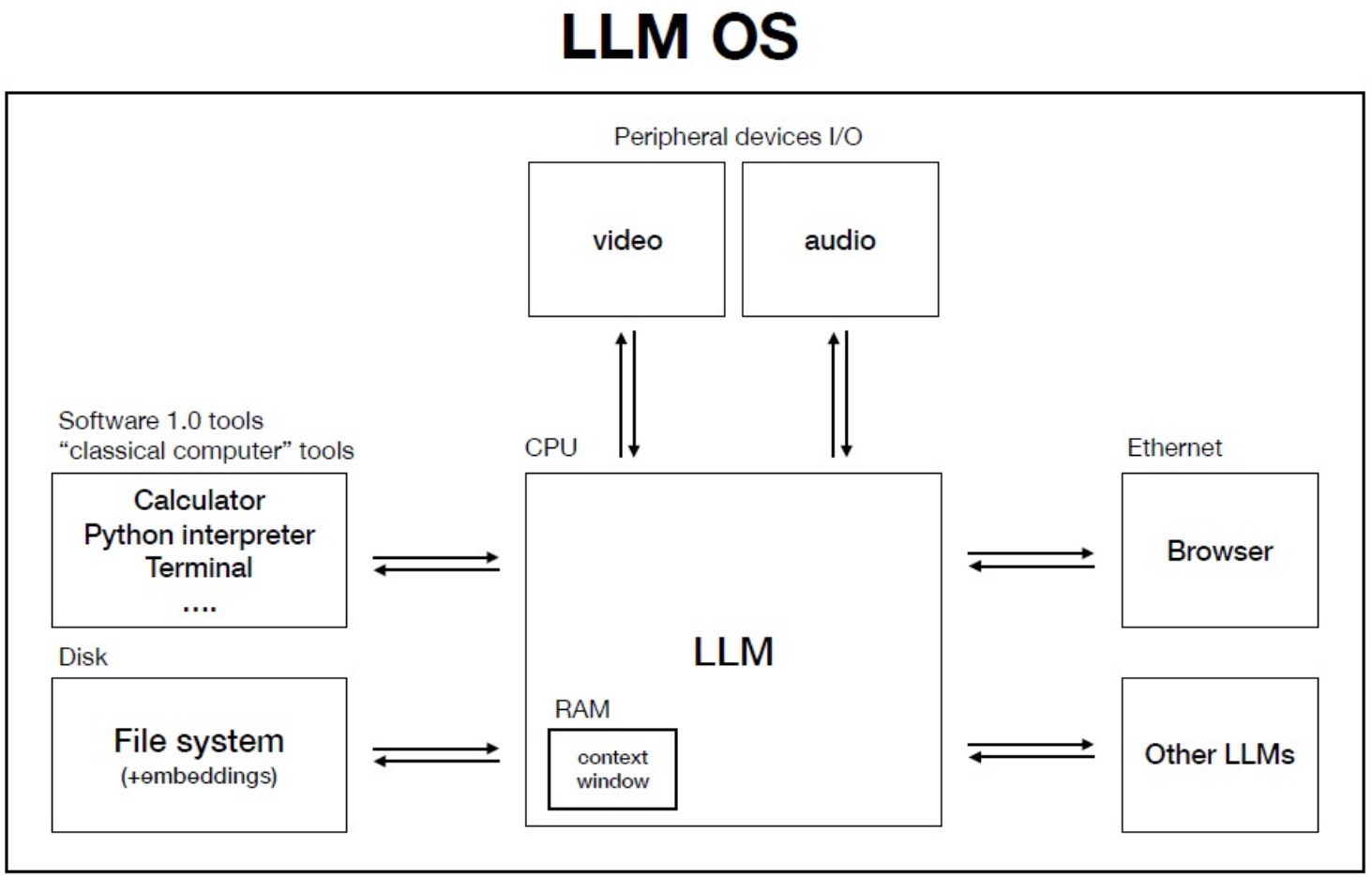
Token Process of LMM for AI Robotics



REF: [50-52]

LLM / LMM Collaboration with Mobile OS and Cloud

Concept of LLM OS

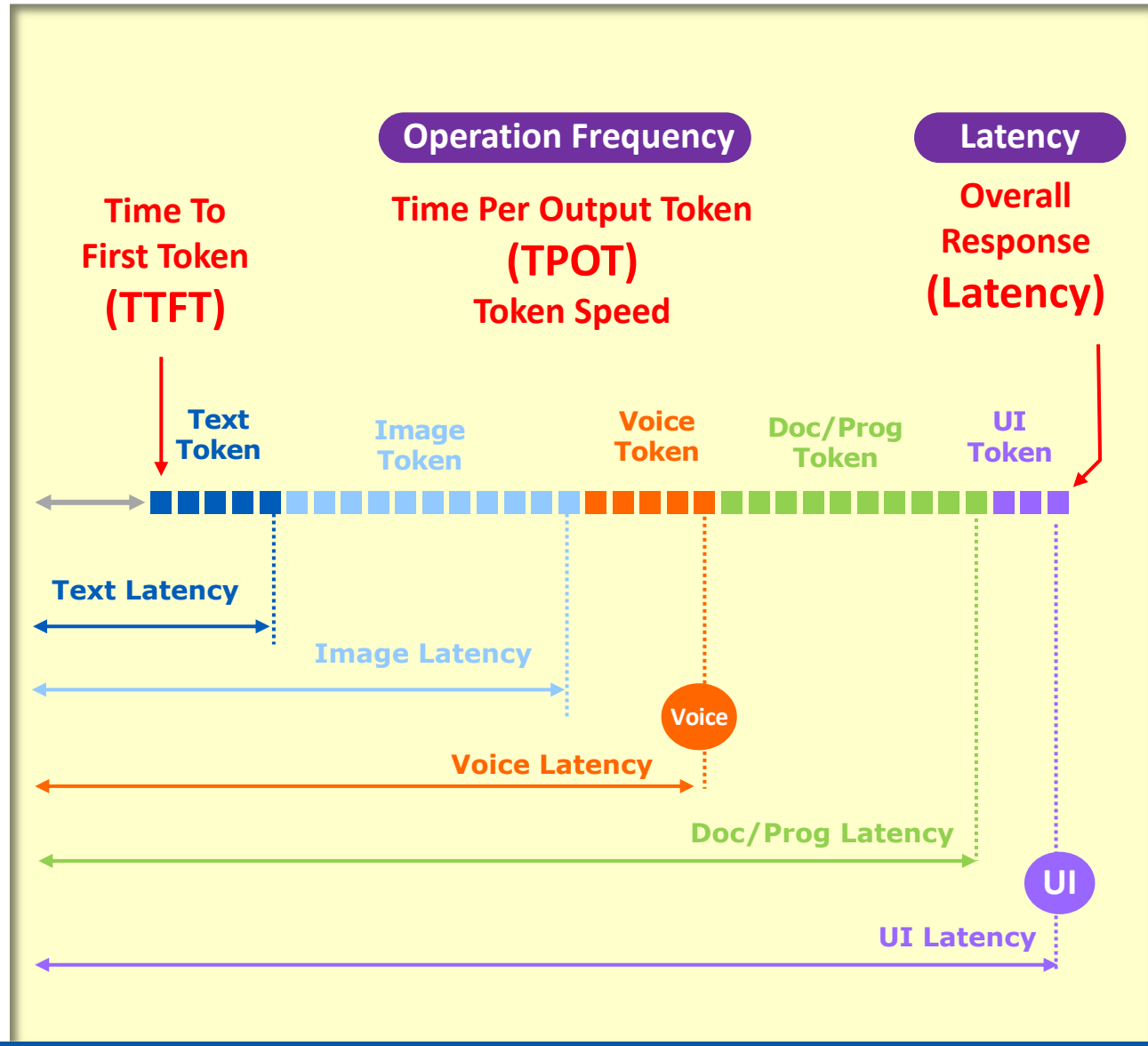


Courtesy : Andrej Karpathy

REF: Karpathy [53]

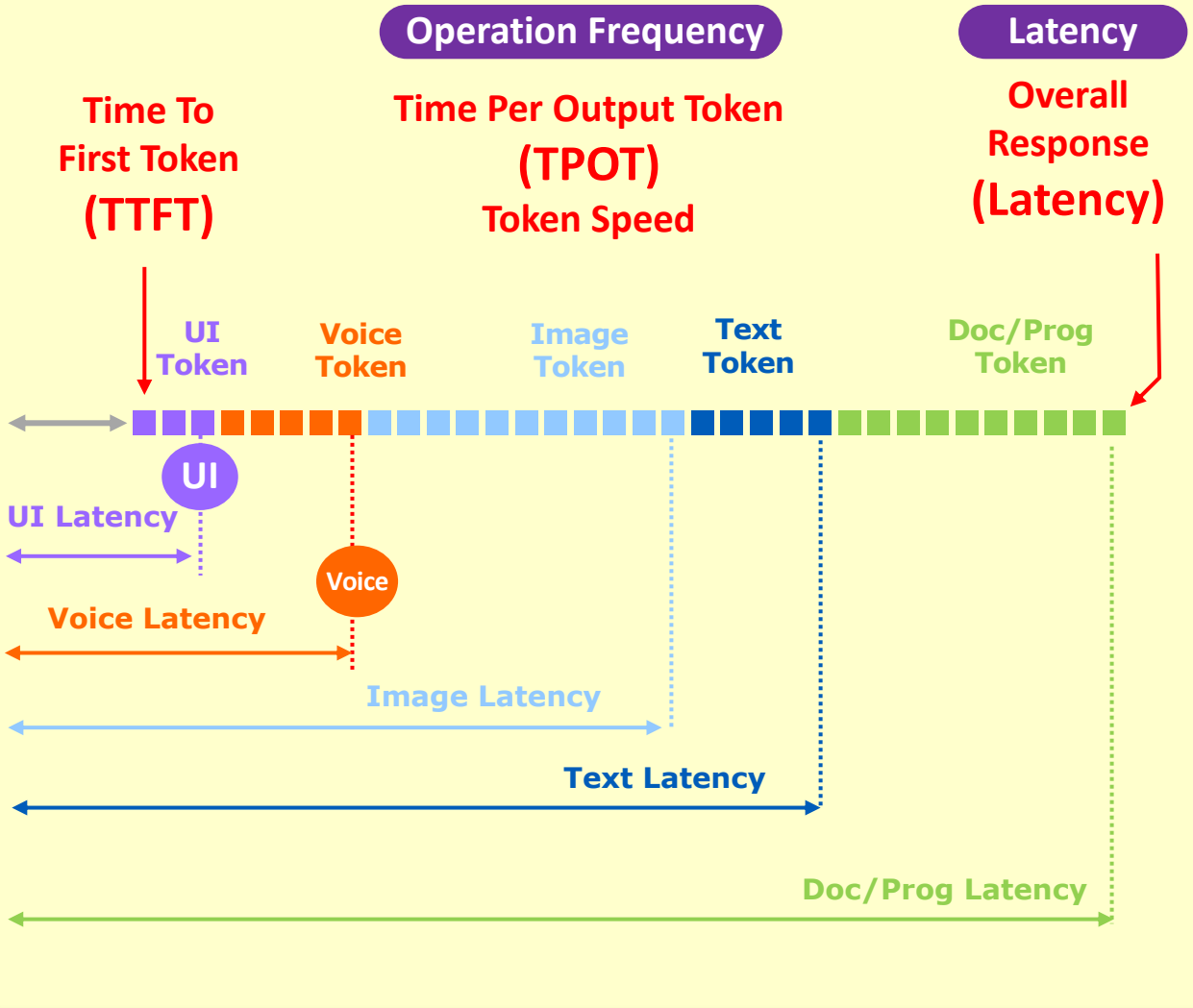
Issues for LLM OS

- **Operation Speed**
  - LLM :~10s token/sec
  - Mobile OS : time measured in ms
    ~1000 Ticks /sec
    (if time interrupt for every 1 ms)
- **Realtime**
  - Overall time to finish a Task
  - Time to finish Sub-tasks
- **Interrupt**
- **Multi-thread / Task Switch**
- **Priority**

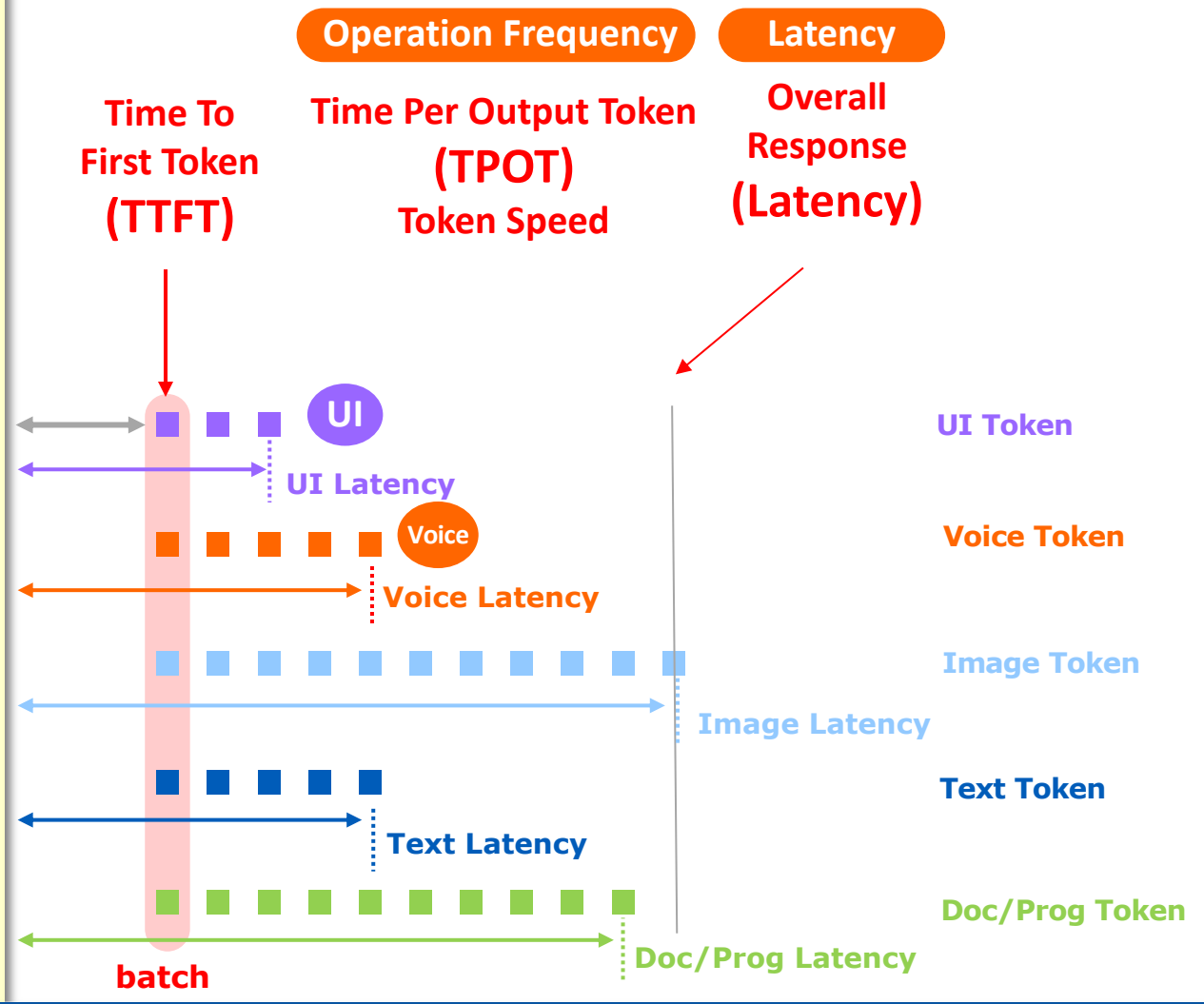


LLM OS : Re-ordering and Parallel Batch

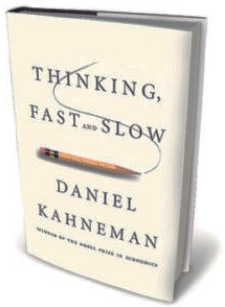
Re-order Sub-tasks in Sequence



Group Independent Sub-tasks into Parallel Batch



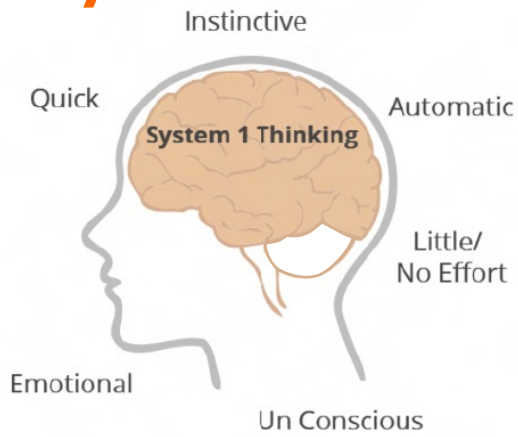
Current LLM/LMM : from "System 1" to "System 2"



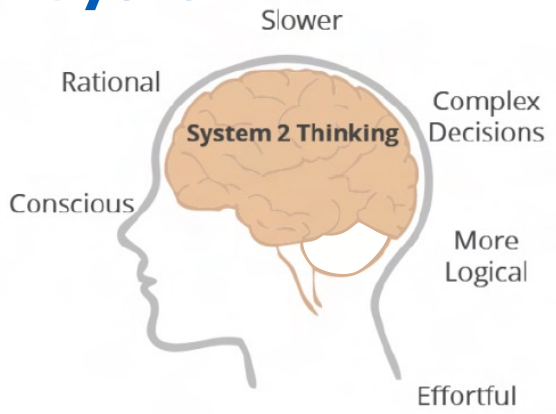
Cerebrum

Current LLM/LMM Status

System 1 : Instinctive



System 2 : Slower



Inspired from System 1 and 2 :

Can we collaborate both Device-side and Cloud-side for System 1 and 2 ?

REF: Karpathy [53], Kahneman [54]

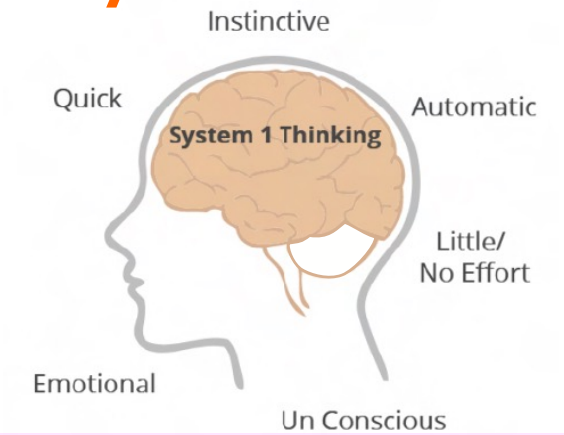
LLM/LMM – Thinking Fast, Deep Thinking and Swift Response

LLM/LMM on Device

LLM/LMM in Cloud

Cerebrum

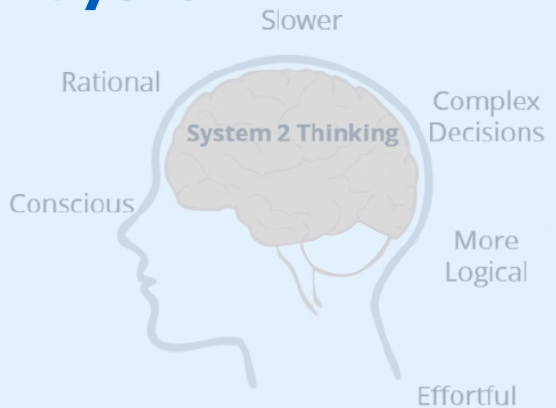
System 1 : Instinctive



Thinking Fast

Fast

System 2 : Slower



Deep Thinking

~~Slower~~ Fast

Cerebellum

Brainstem

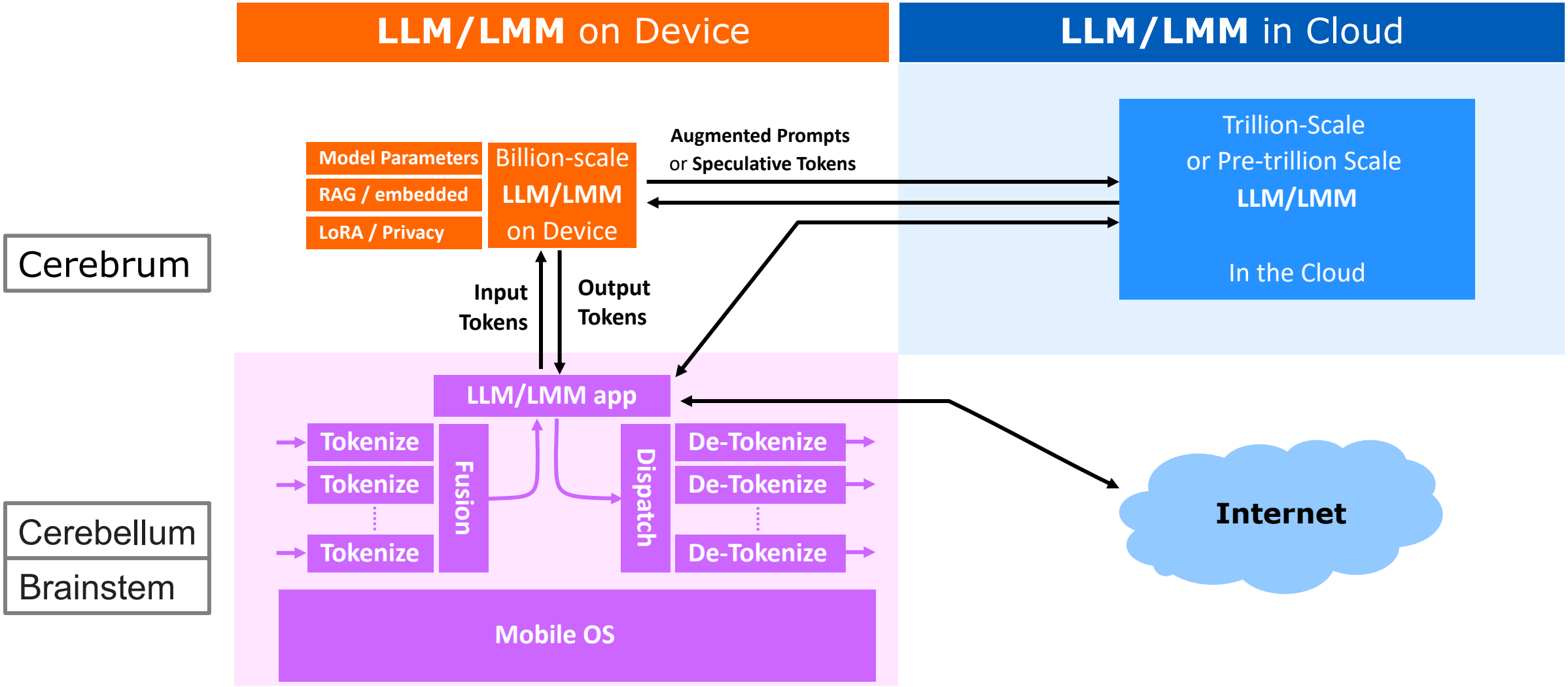
Quick Response
Multimodal Signals
Multi-Tasks Concurrently

Swift Response

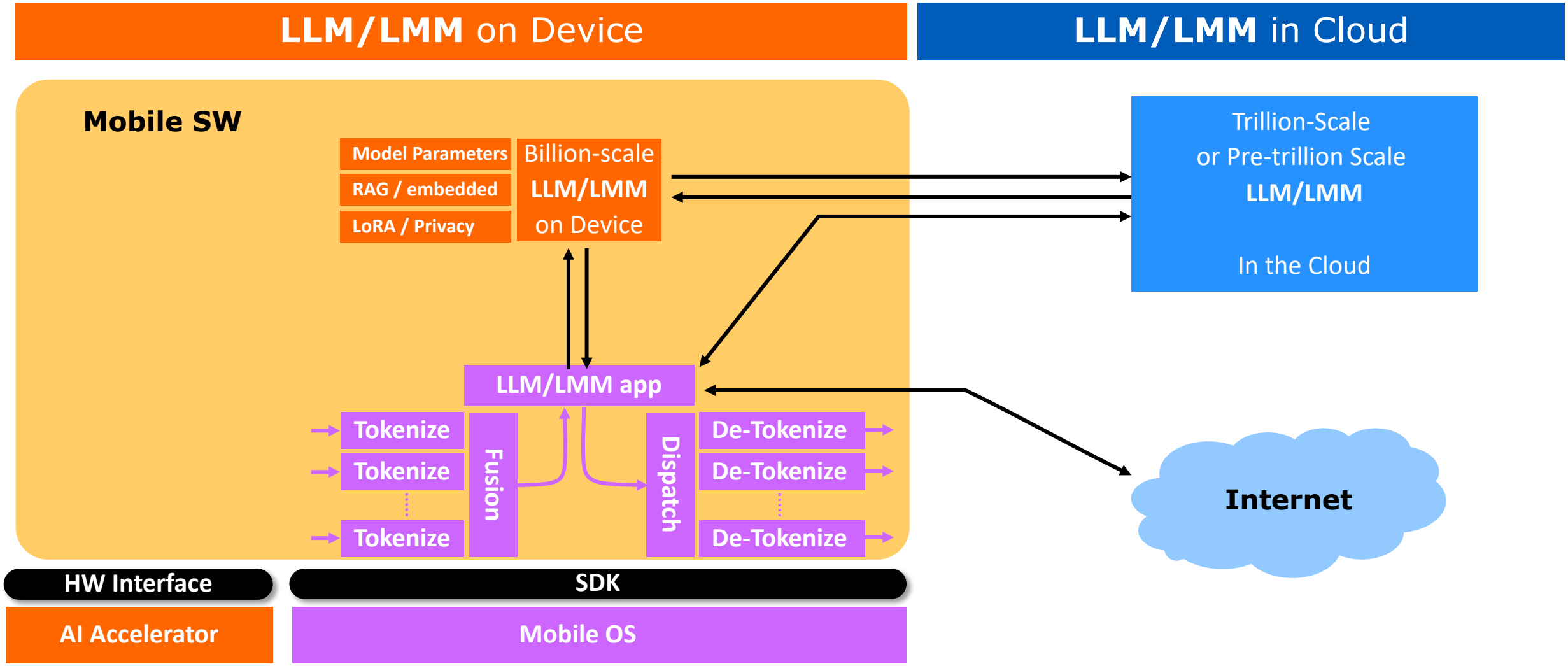
provide sufficient computing resources to speed-up

➔ Cloud Computing

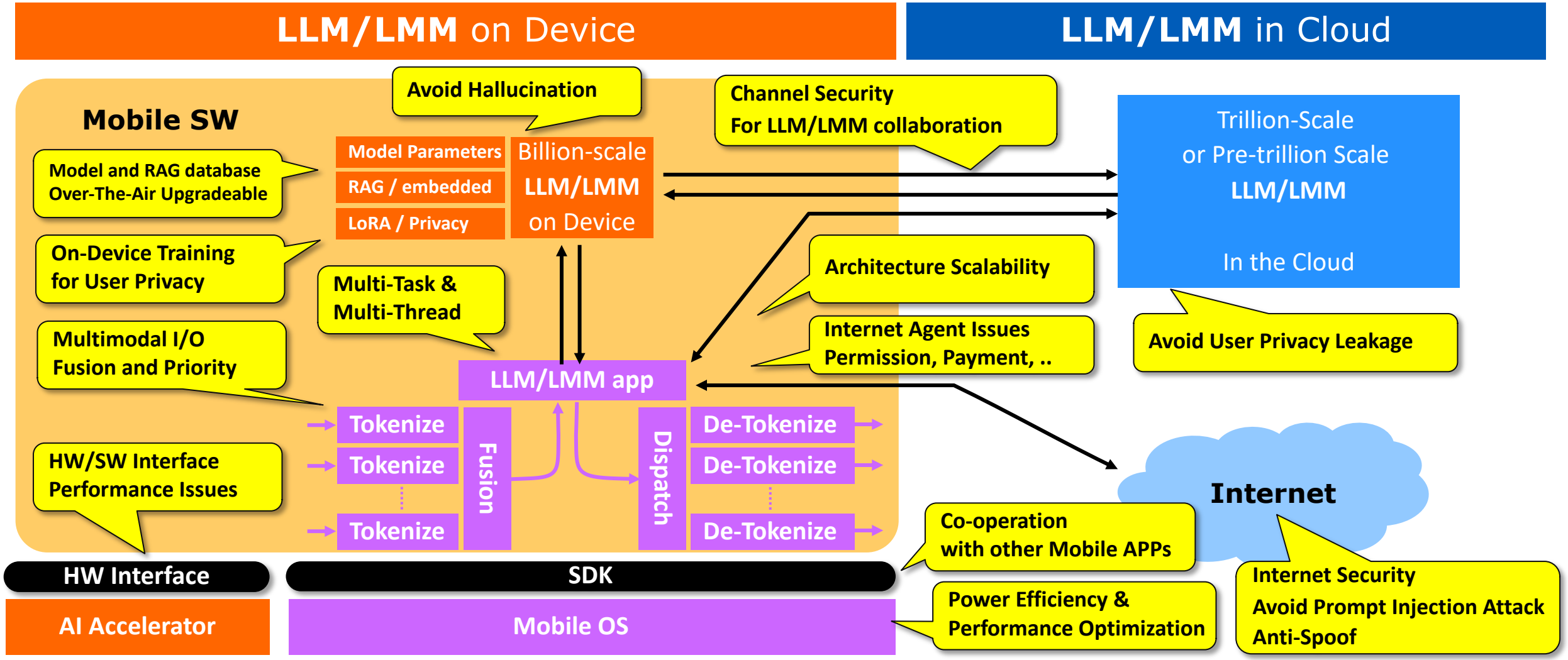
LLM /LMM on Device - Collaborate with Cloud and Mobile OS



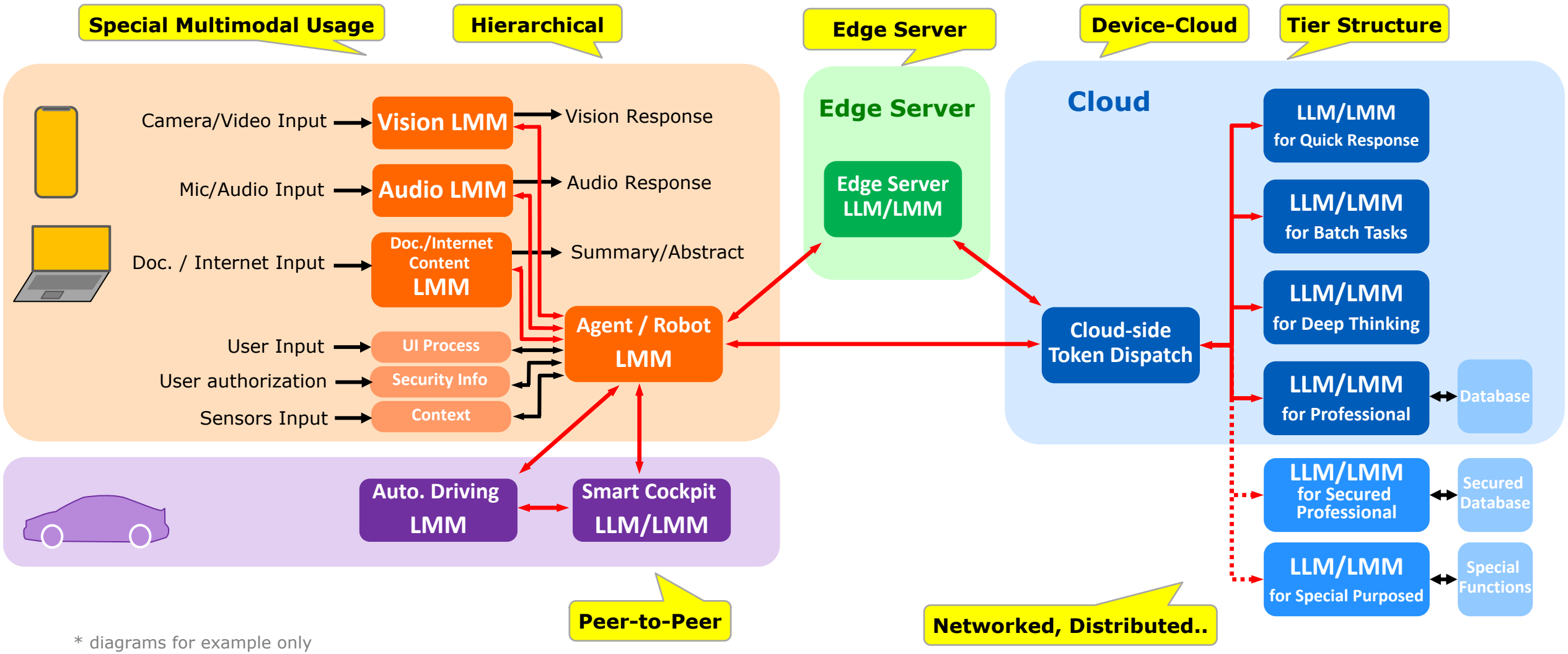
LLM /LMM as Mobile Software on Mobile Processor



LLM /LMM on Mobile Processor : Issues



Collaboration among AI Models



* diagrams for example only

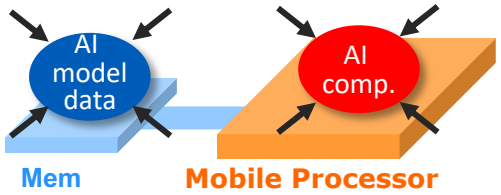
Trends of Mobile Processor Design for LLM/LMM

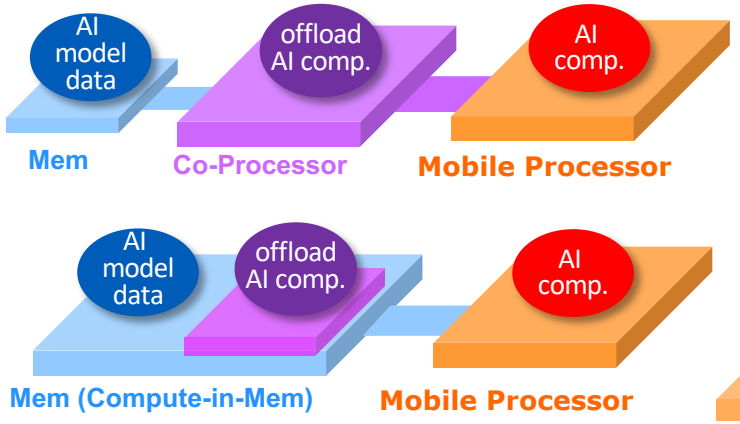
REF: [1-8][31-42][44-53][55-56]

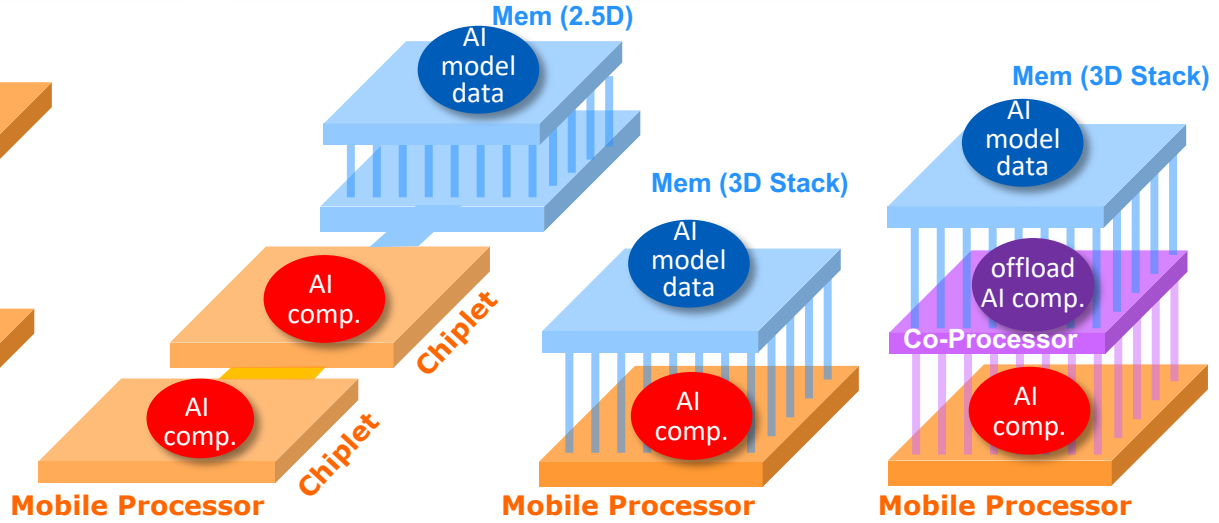
Reduce
Number Representation
Pruning/Sparsity
Smaller model by KD

Offload
Co-processor
Compute in Memory

Scale-up, Scale-out
Chiplets
2.5D/3D Integration





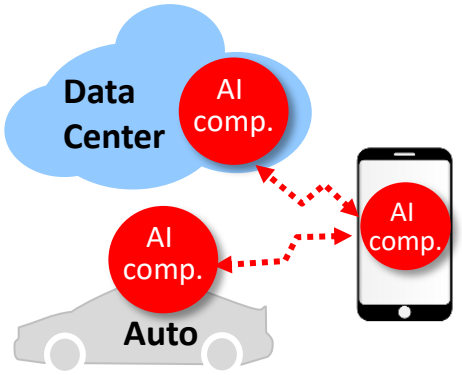


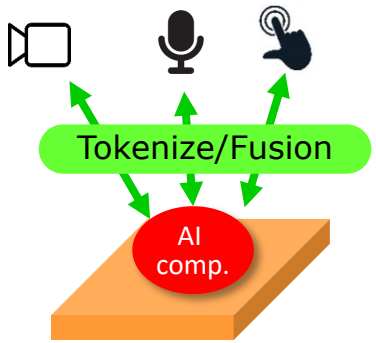
Collaboration

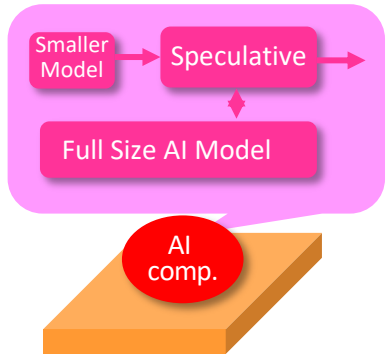
Multimodal
Fusion/Pre-Processing

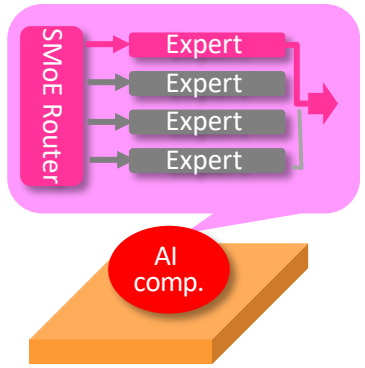
Model/System
Architecture Optimization

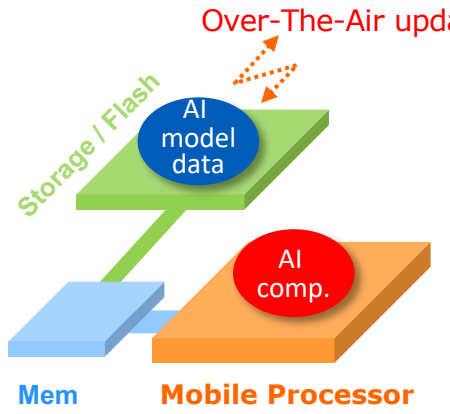
SMoE, RAG, Speculative Execution
LoRA, On-Device Training
Model Management, OTA update











Trends of AI Computing for LLM/LMM - Continuous Evolution

- **AI Model Architecture**
  - **Model Architecture Improvement** (SMoE, RAG, Speculative Execution, LoRA, On-Device Learning..)
  - **Fundamental Model Architecture** (Transformer, Mamba, RWKV..)
  - **Collaboration among Models** (Device-Cloud, Multimodal Partition, Hierarchical, Networked, Distributed..)
- **Hardware Architecture**
  - **2.5D/3D Heterogeneous Integration, Chiplets, Interface** (CoWoS, 3DSoC ; PCIe, UCIe,.)
  - **Non-Von Neuman Architecture** (Computing-in-Memory, Computing-Near-Memory, PIM,..)
  - **Memory Architecture and Emerging Memory** (HBM, CXL,..; ReRAM, MRAM, PCM,..)
  - **Special Computing/Architecture Type** (Neuromorphics, Analog Computing, Approximate computing, ..)
- **Device Types for Applicaitons**
  - **AI Smartphone, AI PC**
  - **Portable AI Assistant** (AI pin, Rabbit R1,..)
  - **AR/VR/XR, Spatial Computing**
  - **Automotive, Robotics**

REF: [1-8][31-42][44-53][55-58]

Conclusion

LLM/LMM to Shape Mobile Processor Design

- **Higher Token Speed :** More AI Accelerators (TOPS) , Memory Bandwidth, Memory Size,..
- **Inference Efficiency :** More Techniques to Improve LLM/LMM Inference on Devices
- **AI Models Collaboration:** Efficient Collaboration among AI Models in Different Devices & Clouds

LLM/LMM to Shape Mobile Device Applications

- **UI by Natural Communication:** Understanding Nuanced Expressions in Natural Language and Body Language
- **AI Agent :** Professional Agents for Device and Internet Interactions
- **Deep Thinking Quality :** Enhanced by Cloud-side LLMs and Networked/Distributed AI Models

AI Inference to Shape LLM/LMM Architecture

- **Optimized AI Model for Inference :** Smaller Model, Specific Architecture, Hardware-Software-Domain Co-design
- **New Fundamental Model Architecture :** New Algorithm and Structure for Higher Efficient Inference

- **An Exciting Area with Profound Impact to Shape Next-Generation Mobile Processors.**
  However, Breakthroughs in Domain-Specific Architecture and in Semiconductor Tech are Required!

References (1/2)

[1] John Hennessy and David Patterson. (2019) “A New Golden Age for Computer Architecture”. *Commun. ACM* 62 (2), Feb 2019, 48-60. https://doi.org/10.1145/3282307

[2] IRDS. (2023) “International Roadmap for Devices and Systems (IRDS) 2023 Update”. *IEEE IRDS (International Roadmap for Devices and Systems).* https://irds.ieee.org/editions/2023

[3] MediaTek. (2023) “MediaTek Dimensity 9300”. https://www.mediatek.com/products/smartphones-2/mediatek-dimensity-9300

[4] Nvidia. (2023) “NVIDIA H200 Tensor Core GPU”. https://www.nvidia.com/en-us/data-center/h200/

[5] AMD. (2023) “AMD Instinct MI300X Platform”. https://www.amd.com/en/products/accelerators/instinct/mi300/platform.html

[6] Google. (2023) “TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embeddings”. *International Symposium on Computer Architecture (ISCA '23)*, 17-21 Jun 2023. https://arxiv.org/abs/2304.01433

[7] Amir Gholami. (2021) “AI and Memory Wall”, 29 Mar 2021. https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8

[8] Dylan Patel and Sophia Wisdom (2023) “On Device AI – Double-Edged Sword”, 13 May 2023. https://www.semianalysis.com/p/on-device-ai-double-edged-sword

[9] Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. (2022) “Compute Trends Across Three Eras of Machine Learning”. 9 Mar 2022. https://arxiv.org/abs/2202.05924

[10] Pablo Villalobos, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, Anson Ho, and Marius Hobbhahn. (2022) “Machine Learning Model Sizes and the Parameter Gap”. 6 Jul 2022. https://arxiv.org/abs/2207.02852

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. (2017) "ImageNet classification with deep convolutional neural networks". *Commun. ACM*. 60 (6), May 2017, 84–90. https://dl.acm.org/doi/10.1145/3065386

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. (2017) “Attention Is All You Need”. 12 Jun 2017. https://arxiv.org/abs/1706.03762

[13] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. (2020) “Scaling Laws for Neural Language Models”, Jan 23, 2020. https://arxiv.org/abs/2001.08361

[14] OpenAI. (2020) “Language Models are Few-Shot Learners”. 28 May 2020. https://arxiv.org/abs/2005.14165

[15] William Fedus, Barret Zoph, and Noam Shazeer. (2021) “Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity”. 11 Jan 2021. https://arxiv.org/abs/2101.03961

[16] Alibaba Group. (2021) “M6-10T: A Sharing-Delinking Paradigm for Efficient Multi-Trillion Parameter Pretraining”. 8 Oct 2021. https://arxiv.org/abs/2110.03888

[17] Samyam Rajbhandari, Olatunji Ruwase, Jeff Rasley, Shaden Smith, and Yuxiong He. (2021) “ZeRO-Infinity: Breaking the GPU Memory Wall for Extreme Scale Deep Learning”. 16 Apr 2021. https://arxiv.org/abs/2104.07857

[18] OpenAI. (2023) “New models and developer products announced at DevDay”. 6 Nov 2023. https://openai.com/blog/new-models-and-developer-products-announced-at-devday

[19] Gemini Team Google. (2023) “Gemini: A Family of Highly Capable Multimodal Models”. 19 Dec 2023. https://arxiv.org/abs/2312.11805

[20] Google Research. (2022) “PaLM: Scaling Language Modeling with Pathways”. 5 Apr. 2022, https://arxiv.org/abs/2204.02311

[21] Meta Research. (2022) “Democratizing access to large-scale language models with OPT-175B”. 3 May 2022. https://ai.meta.com/blog/democratizing-access-to-large-scale-language-models-with-opt-175b/

[22] GenAI, Meta. (2023) “Llama 2: Open Foundation and Fine-Tuned Chat Models”. 18 Jul 2023. https://arxiv.org/abs/2307.09288

[23] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang (2023) “Ferret: Refer and Ground Anything Anywhere at Any Granularity”. 11 Oct 2023. https://arxiv.org/abs/2310.07704

[24] Huawei Technologies. (2023) “PanGu-Σ: Towards Trillion Parameter Language Model with Sparse Heterogeneous Computing”. 20 Mar 2023. https://arxiv.org/abs/2303.10845

[25] The Falcon LLM Team. (2023) “The Falcon Series of Open Language Models”. 28 Nov 2023. https://arxiv.org/abs/2311.16867

[26] Krystal Hu. (2023) “Amazon dedicates team to train ambitious AI model codenamed 'Olympus' -sources”. 8 Nov 2023. https://www.reuters.com/technology/amazon-sets-new-team-trains-ambitious-ai-model-codenamed-olympus-sources-2023-11-08/

[27] Mojan Javaheripi and Sébastien Bubeck. (2023) “Phi-2: The surprising power of small language models”. 12 Dec 2023. https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/

[28] Qwen Team, Alibaba Group. (2023) “Qwen Technical Report”. 28 Sep 2023. https://arxiv.org/abs/2309.16609

[29] Anthropic. (2023) “Claude 2”, 11 Jul 2023. https://www.anthropic.com/news/claude-2

[30] X.ai. (2023) “Announcing Grok”, 4 Nov 2023. https://x.ai/

References (2/2)

[31] Bill Dally. (2023) “Hardware for Deep Learning”. 29 Aug 2023. *Hot Chips* 2023. https://hc2023.hotchips.org/

[32] Jeff Dean and Amin Vahdat. (2023) “Exciting Directions for ML Models and the Implications for Computing Hardware”. 28 Aug 2023. *Hot Chips* 2023. https://hc2023.hotchips.org/

[33] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. (2015) “Distilling the knowledge in a neural network”. 9 Mar 2015. https://arxiv.org/abs/1503.02531

[34] S. Naffziger. (2023) “Innovations For Energy Efficient Generative AI”. 12 Dec 2023. *IEDM* 2023

[35] Megha Agarwal, Asfandyar Qureshi, Nikhil Sardana, Linden Li, Julian Quevedo and Daya Khudia. (2023) "LLM Inference Performance Engineering: Best Practices”. 12 Oct 2023. https://www.databricks.com/blog/llm-inference-performance-engineering-best-practices

[36] Carol chen. (2022) “Transformer Inference Arithmetic”, 20 Mar 2022, https://kipp.ly/transformer-inference-arithmetic/#kv-cache

[37] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela (2020) “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020. https://arxiv.org/abs/2005.11401

[38] Artyom Eliseev, Denis Mazur. (2023) “Fast Inference of Mixture-of-Experts Language Models with Offloading”. 28 Dec 2023, https://arxiv.org/abs/2312.17238

[39] Mistral.AI. (2024) “Mixtral of Experts”, 8 Jan 2024., https://arxiv.org/abs/2401.04088

[40] Andrej Karpathy. (2023), “Speculative execution for LLMs is an excellent inference-time optimization”, Dec 23, 2023. https://twitter.com/karpathy/status/1697318534555336961

[41] Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. (2018) “Blockwise Parallel Decoding for Deep Autoregressive Models”, 7 Nov 2018, https://arxiv.org/abs/1811.03115

[42] Yaniv Leviathan, Matan Kalman, and Yossi Matias (2022) “Fast Inference from Transformers via Speculative Decoding”, 30 Nov 2022, https://arxiv.org/abs/2211.17192

[43] Marc Brysbaert. (2019) “How many words do we read per minute? A review and meta-analysis of reading rate”. Aug. 2019, Journal of Memory and Language 109(104047), DOI:10.1016/j.jml.2019.104047

[44] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang (2023) “The Dawn of LMMs: Preliminary Explorations with GPT-4V(ision)”. 29 Sep 2023. https://arxiv.org/abs/2309.17421

[45] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby (2020) “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. 22 Oct 2020. https://arxiv.org/abs/2010.11929

[46] Zhen Yang, Yingxue Zhang, Fandong Meng, and Jie Zhou, (2023) “TEAL: Tokenize and Embed ALL for Multi-modal Large Language Models”. 8 Nov 2023. https://arxiv.org/abs/2311.04589

[47] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. (2023) “NExT-GPT: Any-to-Any Multimodal LLM”. 11 Sep 2023. https://arxiv.org/abs/2309.05519

[48] An Yan, Zhengyuan Yang, Wanrong Zhu, Kevin Lin, Linjie Li, Jianfeng Wang, Jianwei Yang, Yiwu Zhong, Julian McAuley, Jianfeng Gao, Zicheng Liu, and Lijuan Wang. (2023). “GPT-4V in Wonderland: Large Multimodal Models for Zero-Shot Smartphone GUI Navigation”. 13 Nov 2923, https://arxiv.org/abs/2311.07562

[49] Zane Durante, Qiuyuan Huang, Naoki Wake, Ran Gong, Jae Sung Park, Bidipta Sarkar, Rohan Taori, Yusuke Noda, Demetri Terzopoulos, Yejin Choi, Katsushi Ikeuchi, Hoi Vo, Li Fei-Fei, and Jianfeng Gao. (2024) “Agent AI: Surveying the Horizons of Multimodal Interaction”. 7 Jan 2024. https://arxiv.org/abs/2401.03568

[50] DeepMind. (2022) “A Generalist Agent”. 11 Nov 2022. https://arxiv.org/abs/2205.06175

[51] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. (2023) “PaLM-E: An Embodied Multimodal Language Model”, 6 Mar 2023. https://arxiv.org/abs/2303.03378

[52] Google DeepMind (2023) “RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control”. 28 Jul 2023. https://robotics-transformer2.github.io/assets/rt2.pdf

[53] Andrej Karpathy. (2023) “Intro to Large Language Models”. Nov. 2023, https://www.youtube.com/watch?v=zjkBMFhNj_g

[54] Daniel Kahneman. (2013) *Thinking, Fast and Slow*, 2013, ISBN: 9780606275644h

[55] Gyeong-In Yu, Joo Seong Jeong, Geon-Woo Kim, Soojeong Kim, and Byung-Gon Chun (2022) “Orca: A Distributed Serving System for Transformer-Based Generative Models”, *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)* , https://www.usenix.org/conference/osdi22/presentation/yu

[56] Apple. (2023) “LLM in a flash: Efficient Large Language Model Inference with Limited Memory”. 12 Dec 2023, https://arxiv.org/abs/2312.11514

[57] Albert Gu, Tri Dao. (2023) “Mamba: Linear-Time Sequence Modeling with Selective State Spaces”. 1 Dec 2023. https://arxiv.org/abs/2312.00752

[58] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, et al. (2023) “RWKV: Reinventing RNNs for the Transformer Era”. 11 Dec 2023. https://arxiv.org/abs/2305.13048