Lesson 15

Networking Technologies in the age of Artificial Intelligence

Van-Linh Nguyen

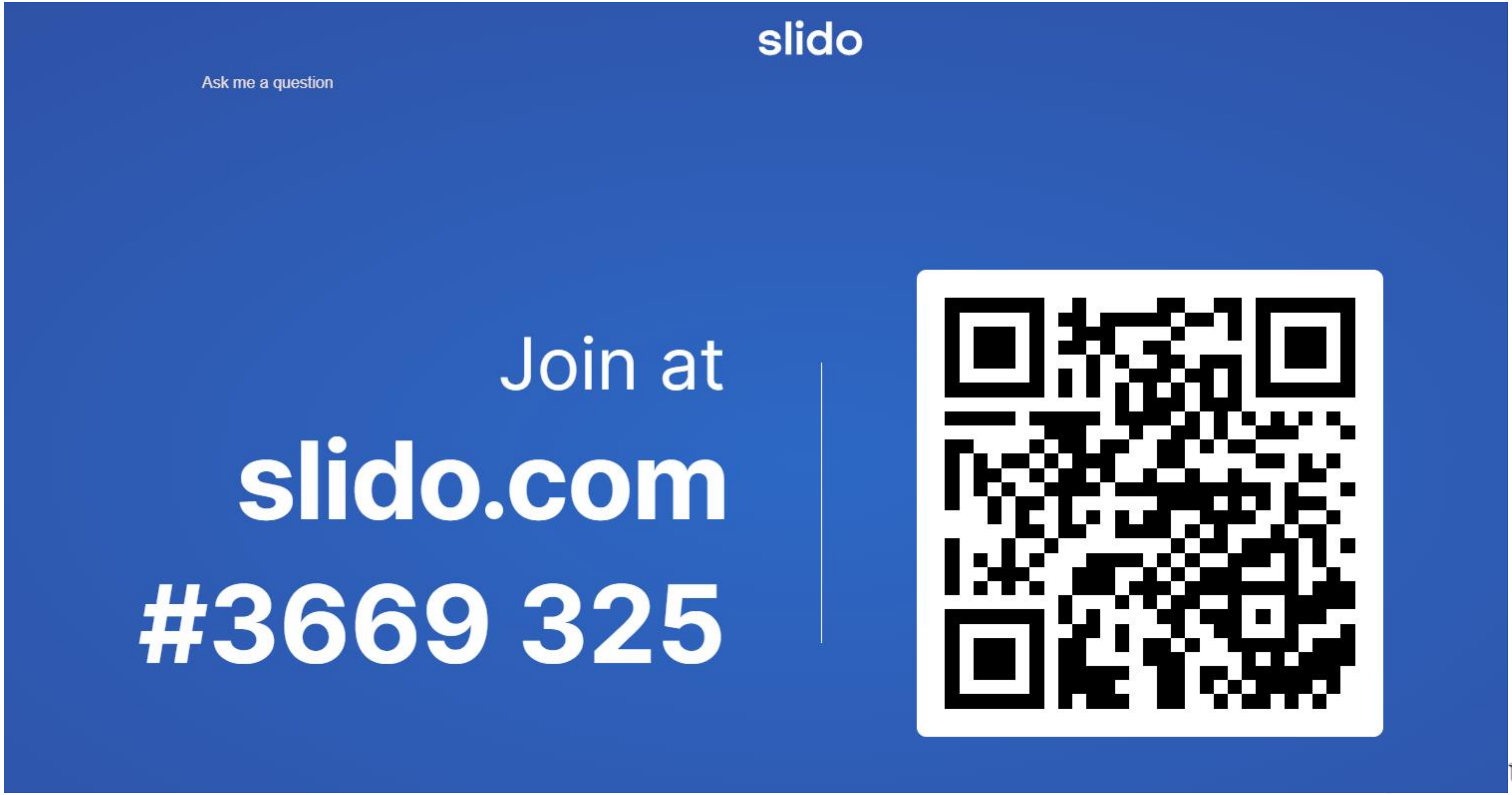
Fall 2024

Outline

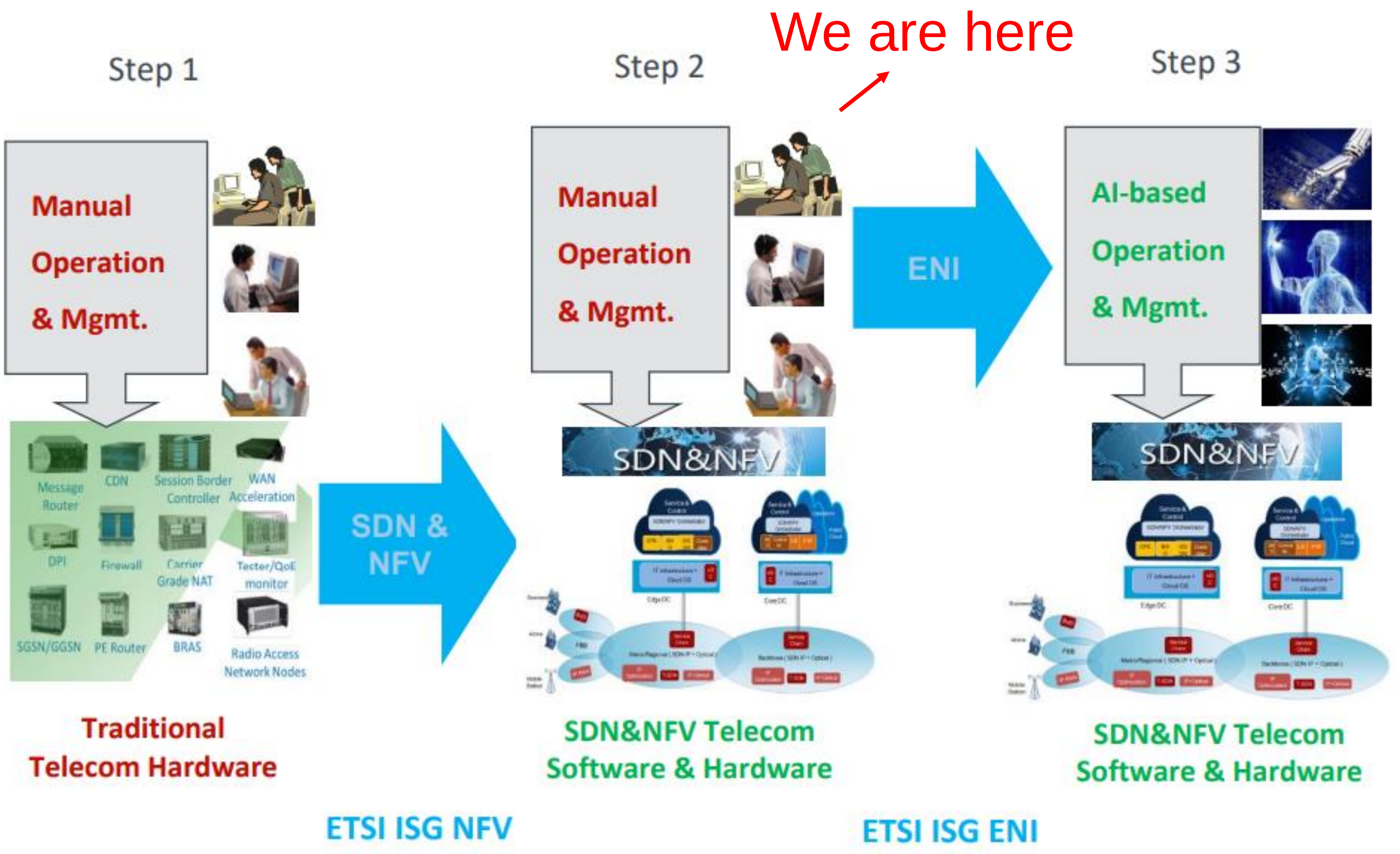
- Future networking technologies
- AI meets Network

Ask me a question without revealing your name

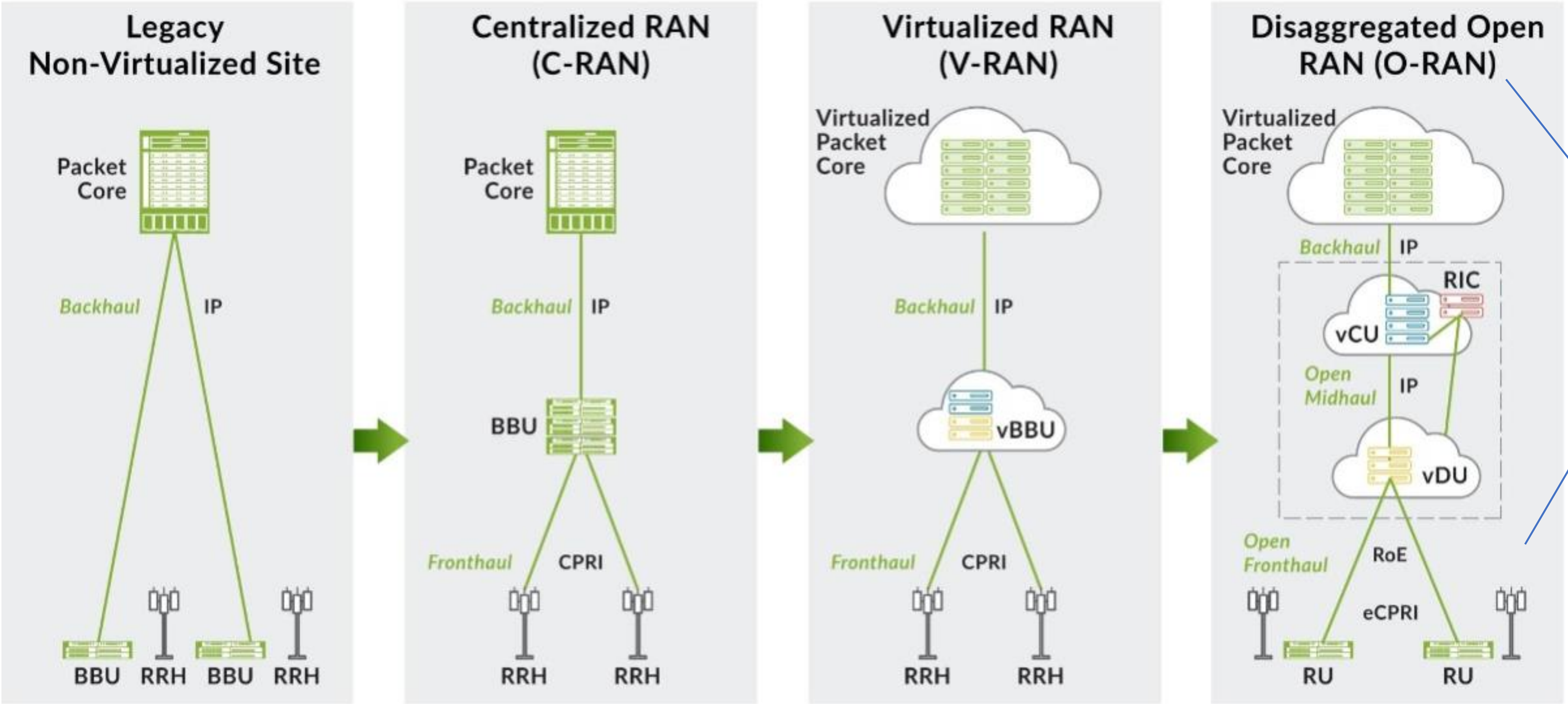
https://app.sli.do/event/esByjf9pUGfaMdFFGhYcpP



How network technologies evolve



From Legacy RAN to Cloud RAN to Open RAN



Develop for 5G/B5G and create a path for 6G

RRH = Remote Radio Head
BBU = Baseband Unit
CPRI = Common Private Radio Interface

RIC = RAN Intelligent Controller
CU = Centralized Unit
DU = Distributed Unit

RU = Remote Unit
RoE = Radio over Ethernet
eCPRI = Ethernet CPRI

https://www.juniper.net/us/en/research-topics/what-is-open-ran.html

Source: OPEN AI CELLULAR

O-RAN Alliance
Work Groups

WG1: Use Cases and Overall Architecture

WG2: Non-RT RIC and A1

WG3: Near-RT RIC and E2

WG4: Open Fronthaul

WG5: Interfaces

WG6: Cloudification and Orchestration

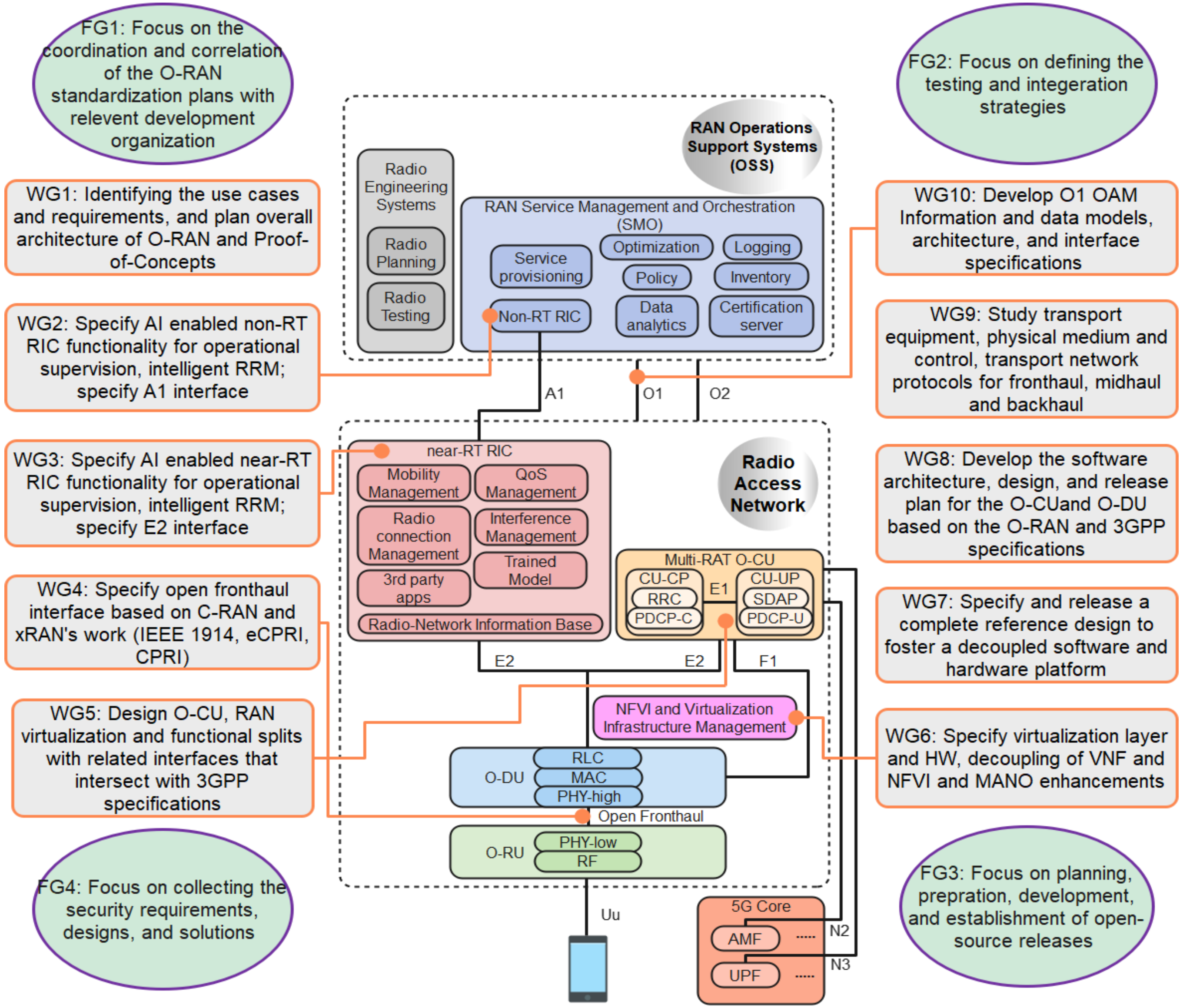
WG7: White-Box Hardware

WG8: Stack Reference Design

WG9: Open X-haul Transport

WG10: OAM

WG11: Security



O-RAN alliance

Contributors (Some)

Operator Members

From Taiwan

Source: https://www.o-ran.org/membership

O-RAN Contributors

From Taiwan

Source: https://www.o-ran.org/membership

Traditional vs Next-Gen Radio Access Networks

- Monolithic
- Inflexible
- Proprietary

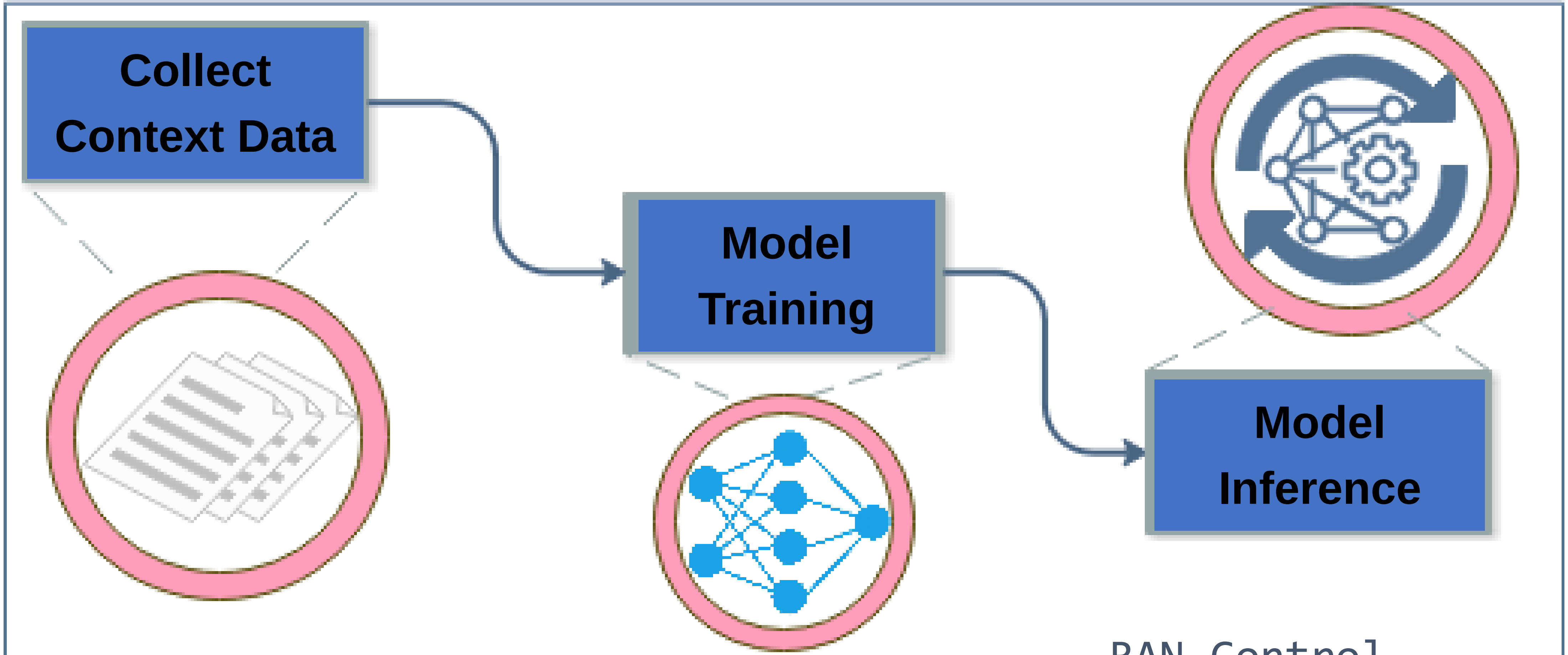
Moving towards softwarization, virtualization…

NFV Servers

- Disaggregated
- Flexible, composable
- Programmable, reconfigurable

Source: OPEN AI CELLULAR

Artificial Intelligence Control (RAN Intelligent Controllers—RICs)



Collect Data, e.g. KPIs from RAN

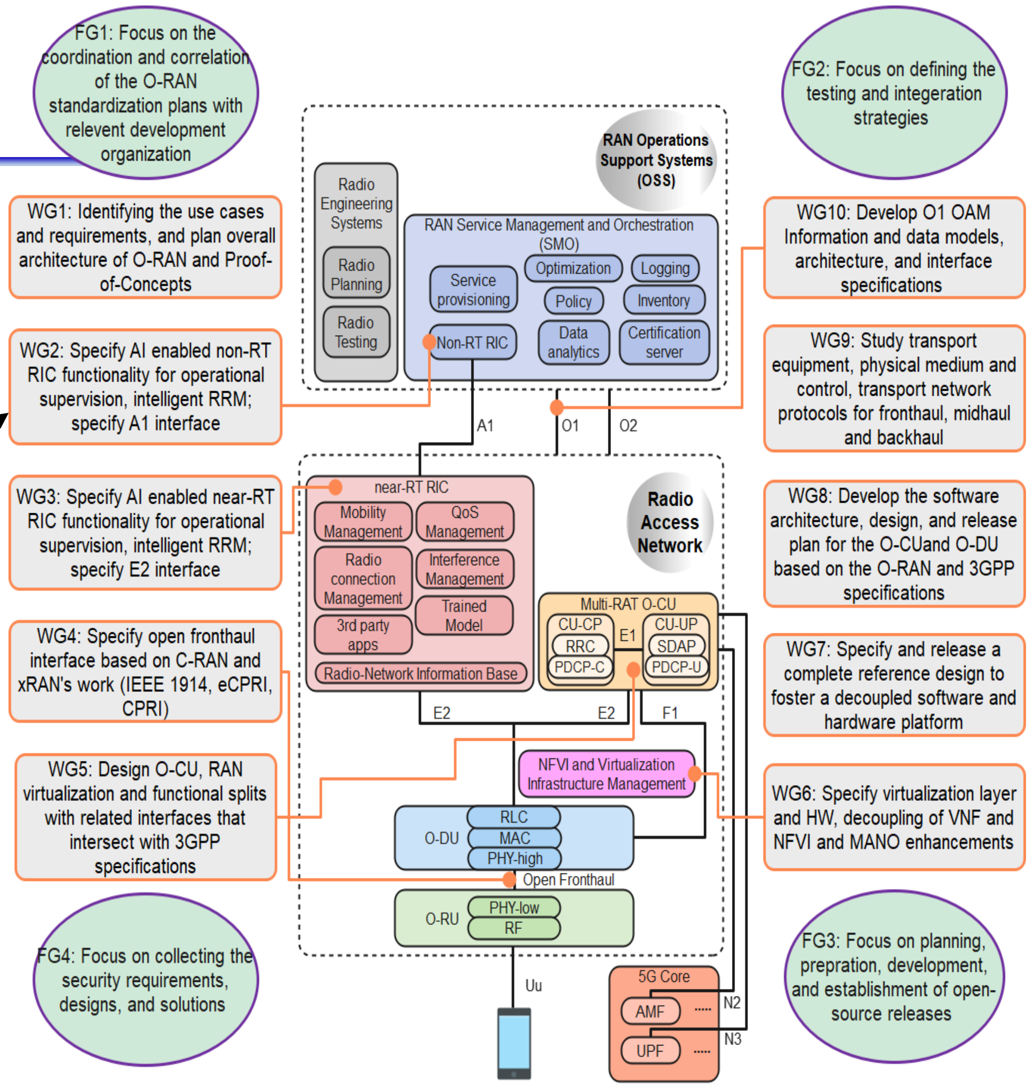
AI Model Training (*non-real time RIC*)

RAN Control
rApps: non-RT RIC
xApps: near-RT RIC

Source: OPEN AI CELLULAR

Where is AI used?

Application	PHY	NET	SERVICE
Radio Resource Allocation (RRA,RRM)	√		
RAN slicing	√	√	√
Mobility Management		√	√
Service offloading		√	√
Load prediction		√	√
Service orchestration	√	√	√
Semantic communications	√	√	√



ersecurity Lab

How to use AI in O-RAN (2)

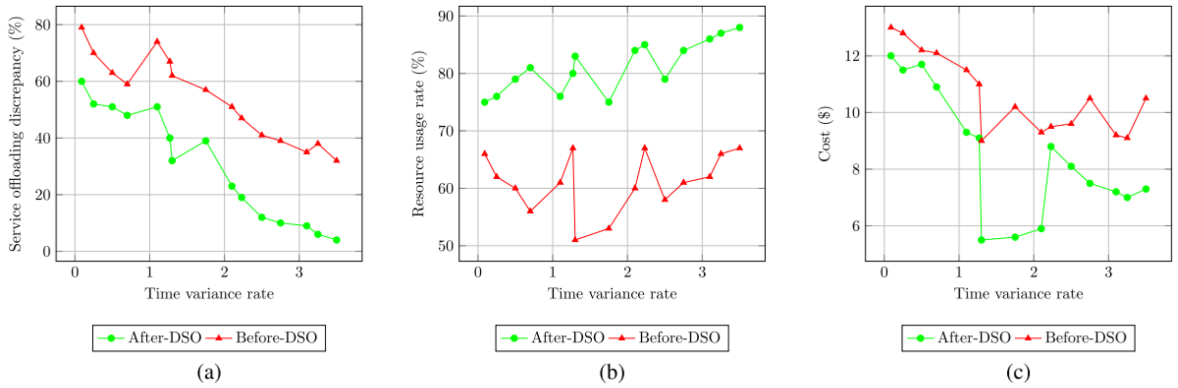
Optimize execution makespan time

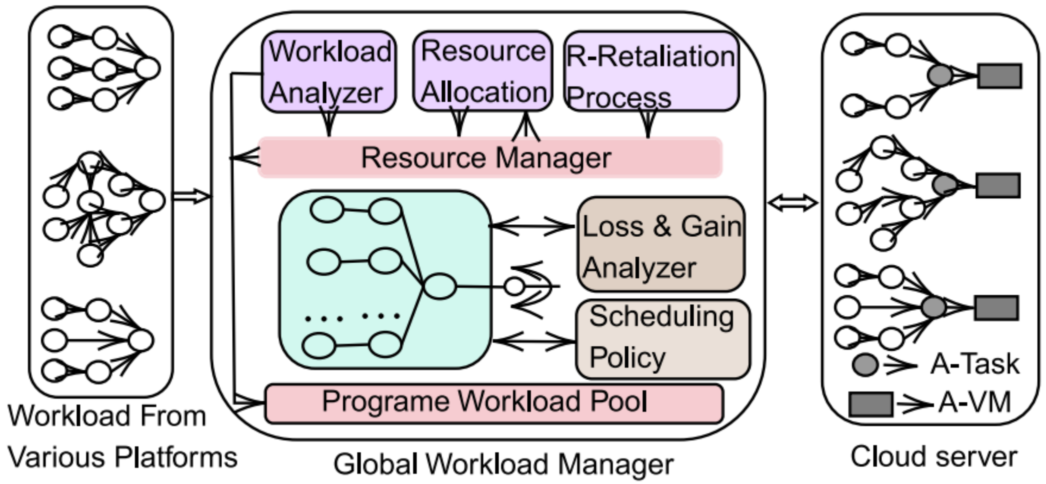
$$\mathrm{MS}_L = \max_{t^j_{i,L} \in T_L} \left\{ \alpha \left(t^j_{\mathrm{ic}} \right) \right\} - \mathrm{AT}_L$$

$$\mathrm{Min} \sum_{L=1}^{F_L} \sum_{j=1}^{\mathrm{ES}_j} \sum_{i=1}^{T_i} \mathbb{Q} \times P_C \times \varpi^j_{i,L}$$

AT is the arrival time, α is the service completion time, $\mathbb{Q}$ is the leased Edge server

- Use **Deep Reinforcement Learning (DRL)** to measure the resource provision rate of the arrived services
- **DRL-based service offloading (DSO) can** reduce subservice execution time cost, transmission time and optimize energy usage rate





1. Mekala et al., A DRL-Based Service Offloading Approach Using DAG for Edge Computational Orchestration, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, 2022
2. Wang et al., Self-play learning strategies for resource assignment in Open-RAN networks, Computer Networks, 2022

How to use AI in O-RAN (3)

$$\max_{P_k,\alpha_{k,r,m}} \sum_{k\in K}\sum_{n\in N_k} w_n r_n$$
$$s.t.\ (1)-(4)$$

where w_n and r_n denotes the priority weight and the reward of each slice

$$r_n^{embb} = \begin{cases} tan^{-1}\left(\sum_{m\in M_n^{embb}} b_m\right), & |H_n^{embb}| \neq \\ 0, & else \end{cases}$$

$$r_n^{urllc} = \begin{cases} 1-\sum_{m\in M_n^{urllc}} d_m, & |H_n^{urllc}| \neq 0 \\ 0, & else \end{cases}$$

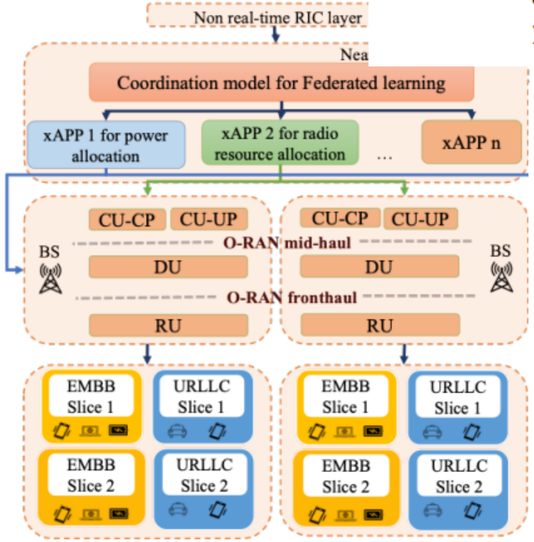
$$P_{min} \le P_k \le P_{max}, \forall k$$
$$\alpha_{k,r,m} = \{0,1\}, \forall k, r, m$$
$$\Sigma_{m\in M}\alpha_{k,r,m} = 1, \forall k, r$$

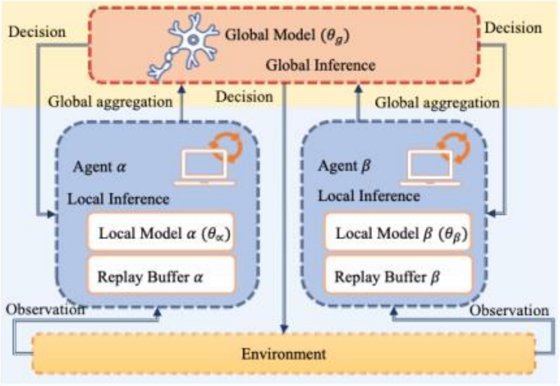
For eMBB slices, optimization aims to achieve maximum total throughput, and for URLLC slices, the aim is to minimize the average delay of packets

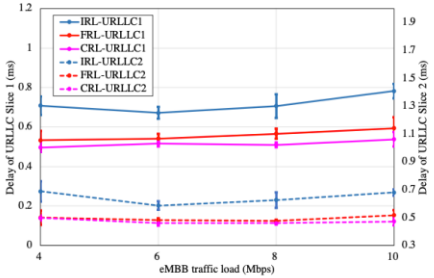
- Use **Deep Reinforcement Learning (DRL)** to control Power and allocate Radio/computing Resource in xAPP Agent

1. **State** (queue length of packets, the current delay and the current transmission power)
2. **Action** (Choose power level, the portion of **RBs** allocated to each slice)
2. **Reward** (the weighted sum reward of all the slices)

$$r_{k,t} = \sum_{n\in N_k} w_n r_n,$$







1. Zhang et al., Federated Deep Reinforcement Learning for Resource Allocation in O-RAN Slicing, GLOBECOM, 2022

國立中正大學 Cybersecurity Lab

Existing AI-based xAPPs

- There are currently several AI-based xAPPs that can use to test the attack performance

Existing xApps

1. Hello World - O-RAN Software Community
2. Bouncer – O-RAN Software Community
3. KPIMON – O-RAN Software Community
4. Traffic Steering – O-RAN Software Com.
5. Load Prediction – O-RAN Software Com.
6. NexRAN (RAN Slicing) – POWDER

New xApps to be released soon

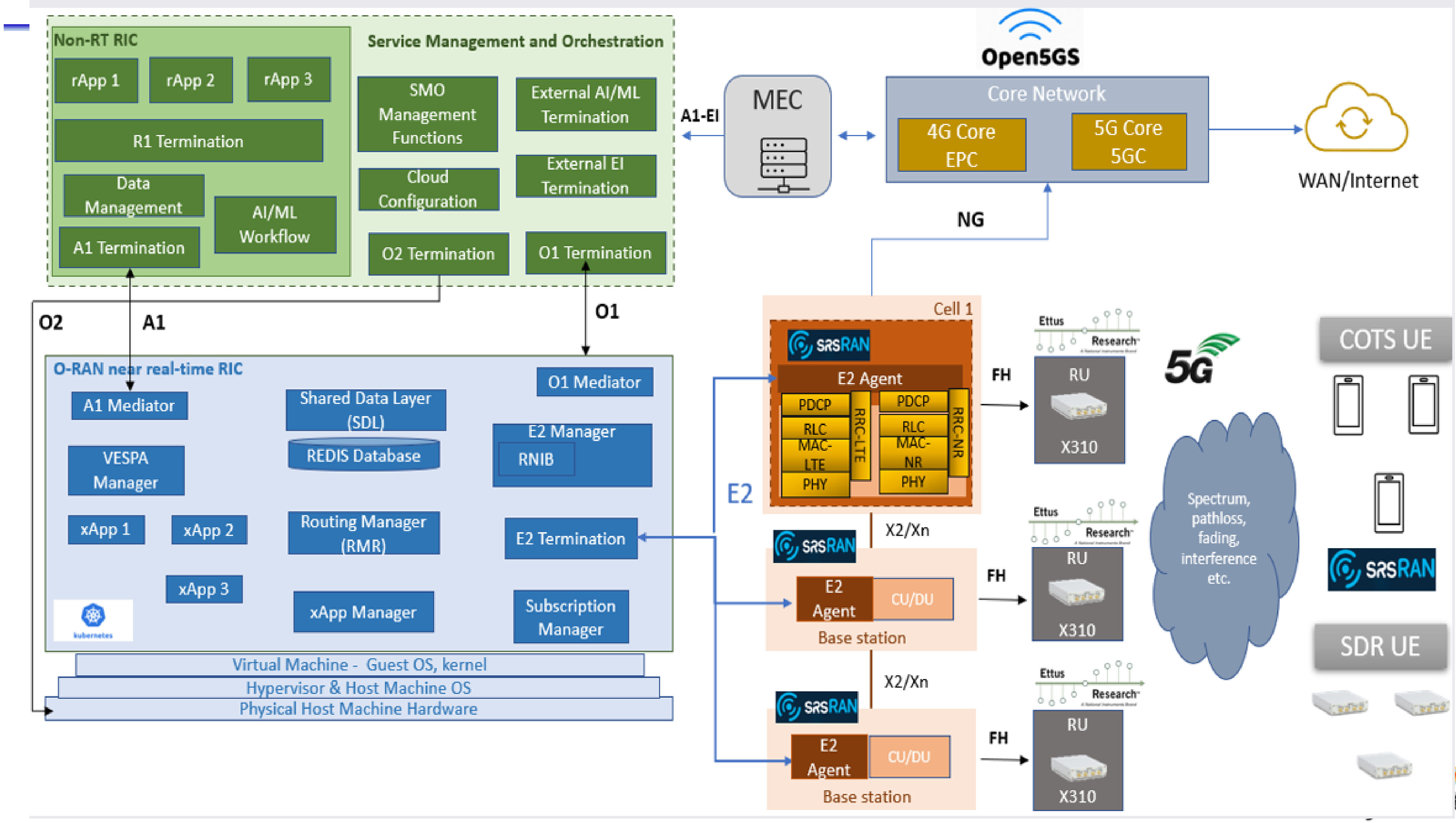
1. Age of Information (AoI) Scheduler
2. AI-Enhanced Schedulers
3. RAN Slicing v2
4. Other xApps

Source: https://www.openaicellular.org/about

B. Tang, et al. "AI Testing Framework for Next-G O-RAN Networks: Requirements, Design, and Research Opportunities," IEEE Wireless, 2023, https://arxiv.org/pdf/2211.03979.pdf

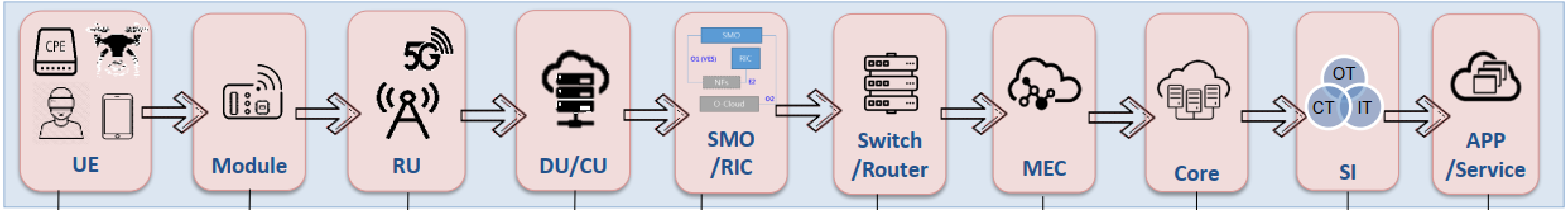
O-RAN testbed

OAIC framework



Taiwan 5G Open Network Ecosystem

End to End Open Network Solution; Your Trusted Agile Partners



HTC, Pegatron,
Askey, ASUS, WNC,
Arcadyan, Comapl,
Gemtek, Mitrastar,
Alpha networks,
…

YTTEK,
TMYTEK,
Auden,
M.gear,
…

Foxconn,
WNC,
MTI,
Synergy,
Lions,
Pegatron,
Compal,
Delta,
Liteon,
Metanoia,
Alpha,
…

QCT,
Foxconn,
Pegatron,
HTC, Alpha,
Compal,
MiTAC,
…

QCT, HTC, Compal,
Pegatron, WNC,
Quanta,
…

Foxconn,
Alpha, Delta,
Edgecore,
UfiSpace,
Wistron,
CyberTAN,
…

ChungHwa
Telecom,
Pegatron,
Compal,
QCT, …

Atayalan,
Foxconn,
Saviah,
QCT,
…

Server

QCT, Compal, Foxconn,
Inventec, Wiwynn,
Pegatron, Advantech,
ADLINK, …

ChungHwa
Telecom,
Fareastone,
Taiwan
Mobile, …

HwaCom, Wave-
In, FHNET,
TransNET, LEO,
Compal, QCT,
Pegatron,
…

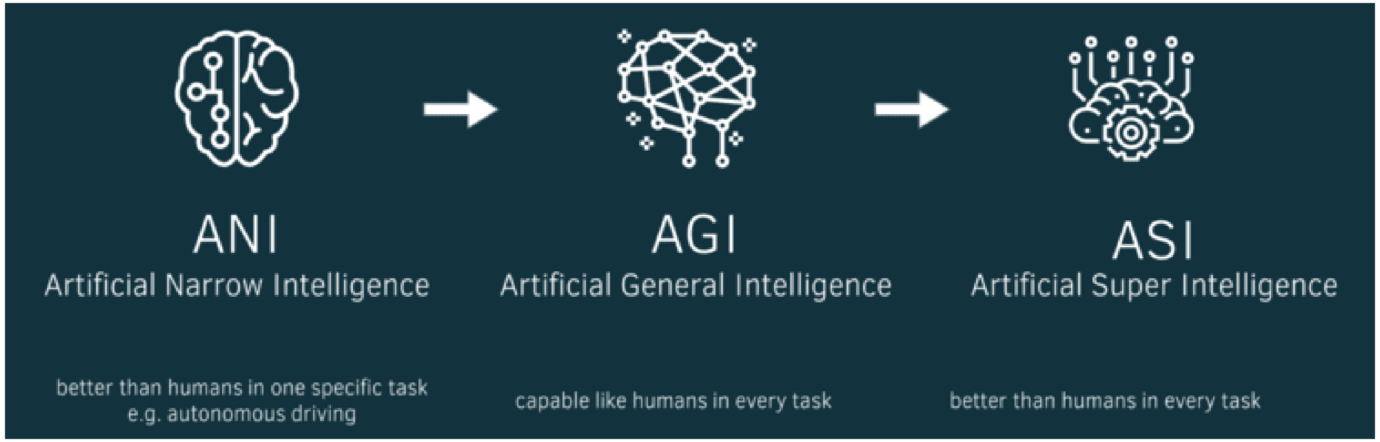
Some futuristic technologies for computer networks

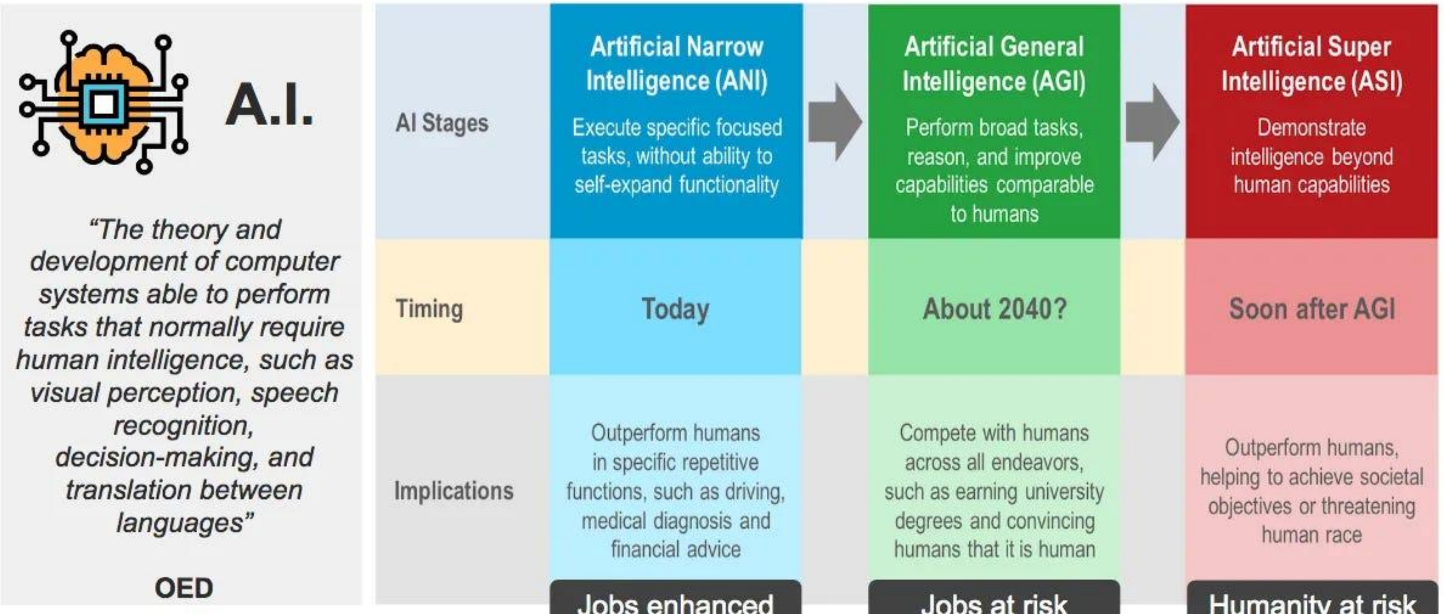
AGI, Foundation Models, Federated Learning

國立中正大學 Cybersecurity Lab

Artificial general intelligence

Src: Cameo



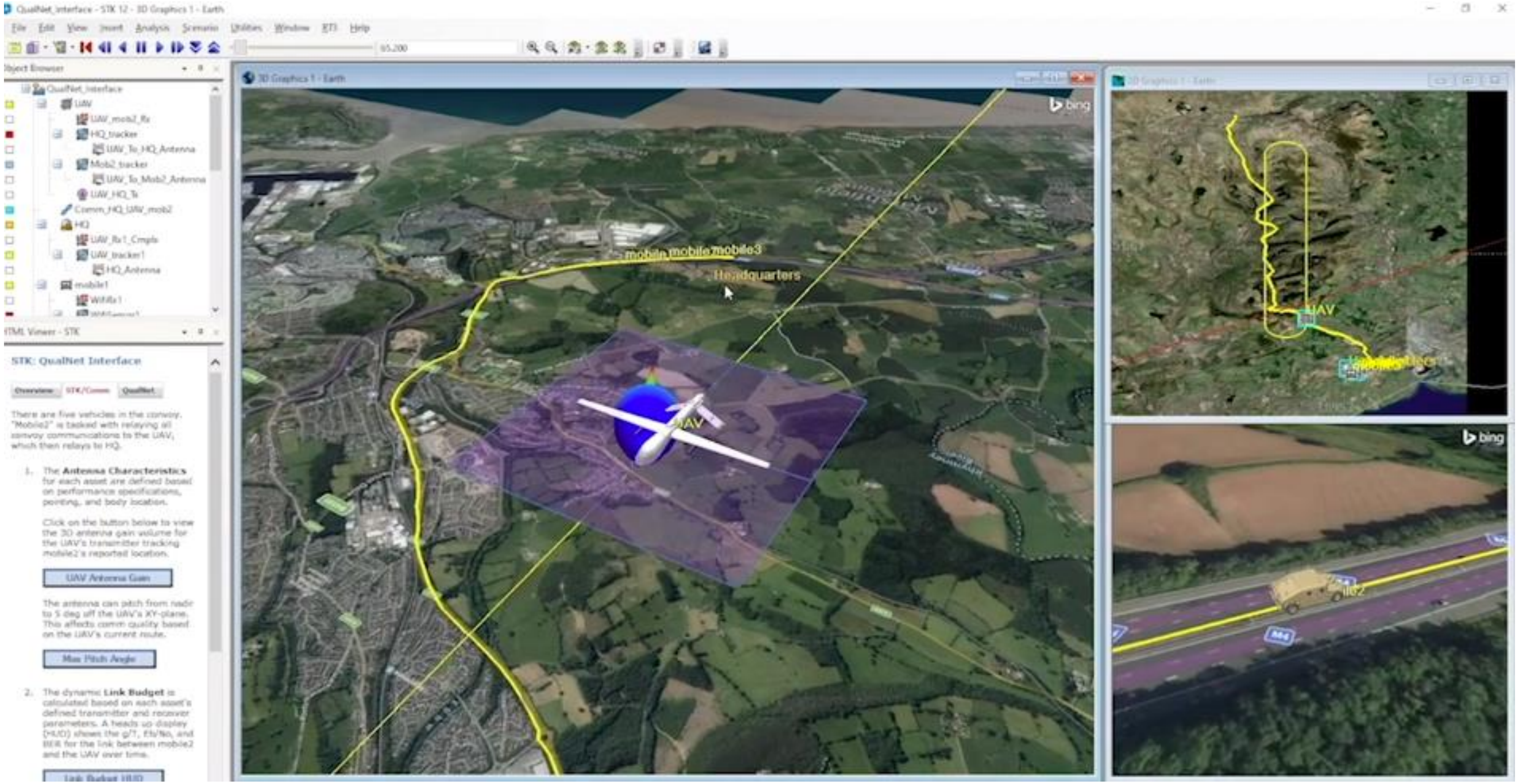


AGI for autonomous robots/vehicles

Src: Alliance For Automotive Innovation

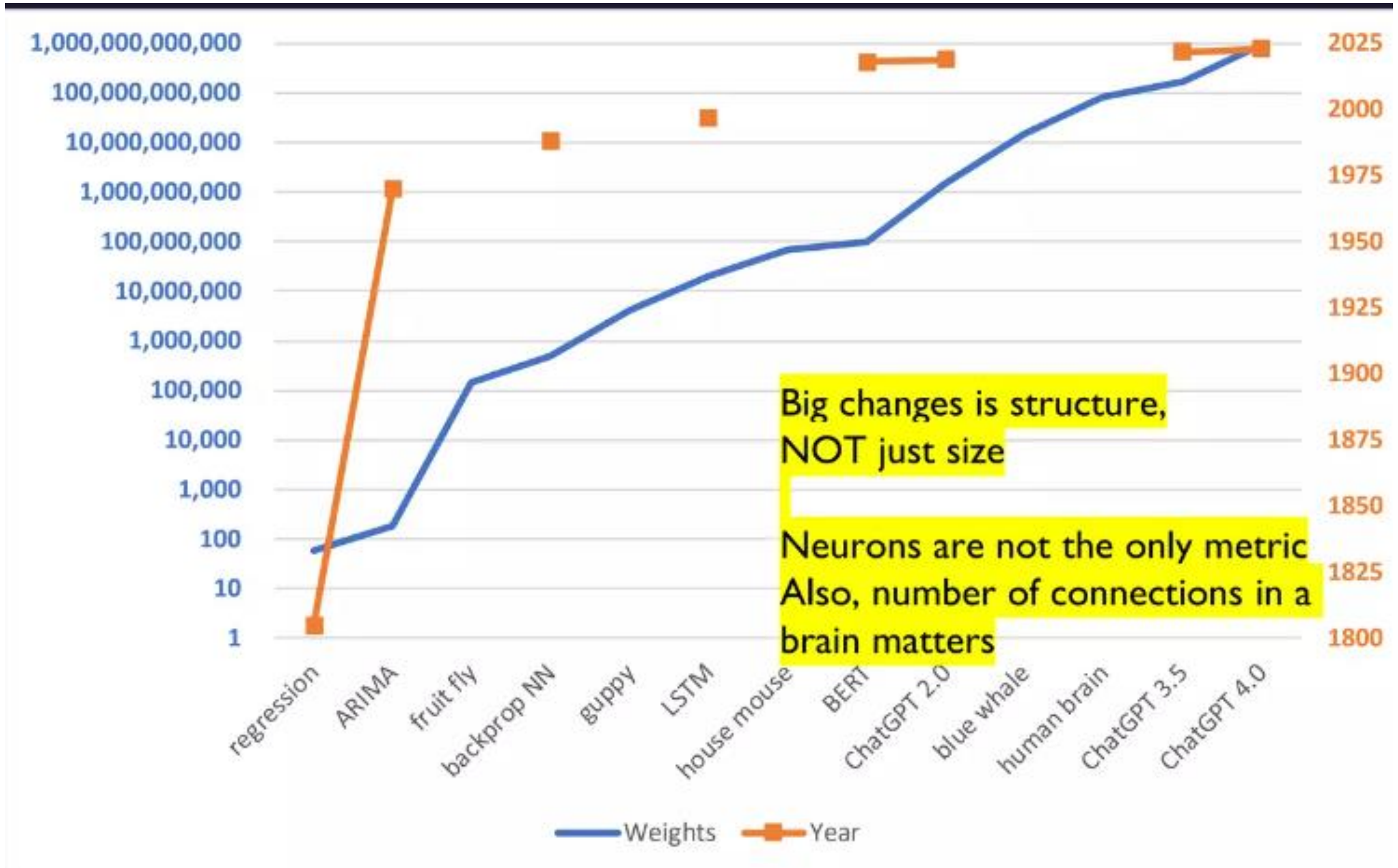
國立中正大學 Cybersecurity Lab

AGI for swarm robots control



國立中正大學 Cybersecurity Lab

How AGI grows



Algorithm	Weights	Year
ChatGPT 4.0	1,000,000,000,000	2023
ChatGPT 3.5	175,000,000,000	2022
human brain	86,000,000,000	
blue whale	15,000,000,000	
ChatGPT 2.0	1,500,000,000	2019
BERT	100,000,000	2018
house mouse	71,000,000	
LSTM	20,000,000	1997
guppy	4,300,000	
backprop NN	500,000	1988
fruit fly	150,000	
ARIMA	180	1970
regression	60	1805

AGI for Rich Guys

Model	Parameters	Jump Factor	Chinchilla Tokens (B)	Jump Factor	CS-2 Config	Days To Train	Jump Factor	Price To Train	Jump Factor	Cost Per 1M Parameters
GPT-3XL	1.3		26		4 * CS-2	0.4		$2,500		$1.92
GPT-J	6	4.6 X	120	4.6 X	4 * CS-2	8	20.0 X	$45,000	18.0 X	$7.50
GPT-3 6.7B	6.7	1.1 X	134	1.1 X	4 * CS-2	11	1.4 X	$40,000	0.9 X	$5.97
T-5 11B	11	1.6 X	*34*	0.3 X	4 * CS-2	9	0.8 X	$60,000	1.5 X	$5.45
GPT-3 13B	13	1.2 X	260	7.6 X	4 * CS-2	39	4.3 X	$150,000	2.5 X	$11.54
GPT NeoX	20	1.5 X	400	1.5 X	4 * CS-2	47	1.2 X	$525,000	3.5 X	$26.25
GPT NeoX	*20*	*1.5 X*	***400***	*1.5 X*	***16 * CS-2***	***11.1***	*0.3 X*	***$656,250***	*4.4 X*	***$32.81***
GPT 70B	70	3.5 X	1,400	3.5 X	4 * CS-2	85	1.8 X	$2,500,000	4.8 X	$35.71
GPT 70B	*70*	*3.5 X*	***1,400***	*3.5 X*	***16 * CS-2***	***21.3***	*0.3 X*	***$3,125,000***	*6.0 X*	***$44.64***
GPT 175B	175	2.5 X	3,500	2.5 X	4 * CS-2	***110.5***	1.3 X	***$8,750,000***	3.5 X	***$50.00***
GPT 175B	*175*	*2.5 X*	***3,500***	*2.5 X*	***16 * CS-2***	***27.6***	*0.3 X*	***$10,937,500***	*4.4 X*	***$62.50***

https://www.nextplatform.com/2022/12/01/counting-the-cost-of-training-large-language-models/

國立中正大學 Cybersecurity Lab

AGI for Rich Guys

	GPU Type	GPU Power consumption	GPU-hours	Total power consumption	Carbon emitted (tCO_2eq)
OPT-175B	A100-80GB	400W	809,472	356 MWh	137
BLOOM-175B	A100-80GB	400W	1,082,880	475 MWh	183
LLaMA-7B	A100-80GB	400W	82,432	36 MWh	14
LLaMA-13B	A100-80GB	400W	135,168	59 MWh	23
LLaMA-33B	A100-80GB	400W	530,432	233 MWh	90
LLaMA-65B	A100-80GB	400W	1,022,362	449 MWh	173

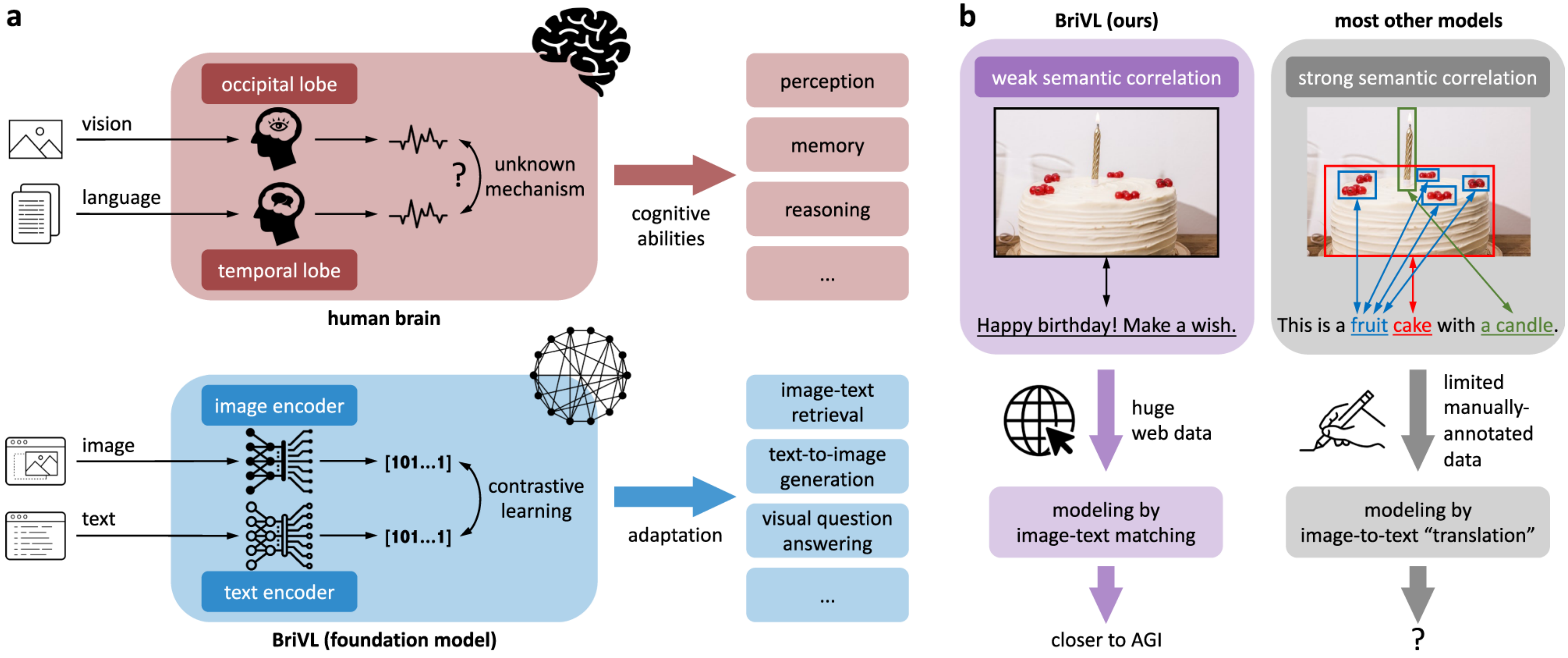
Table 15: **Carbon footprint of training different models in the same data center.** We follow Wu et al. (2022) to compute carbon emission of training OPT, BLOOM and our models in the same data center. For the power consumption of a A100-80GB, we take the thermal design power for NVLink systems, that is 400W. We take a PUE of 1.1 and a carbon intensity factor set at the national US average of 0.385 kg CO_2e per KWh.

https://simonwillison.net/2023/Mar/17/beat-chatgpt-in-a-browser/

AGI and what we should learn

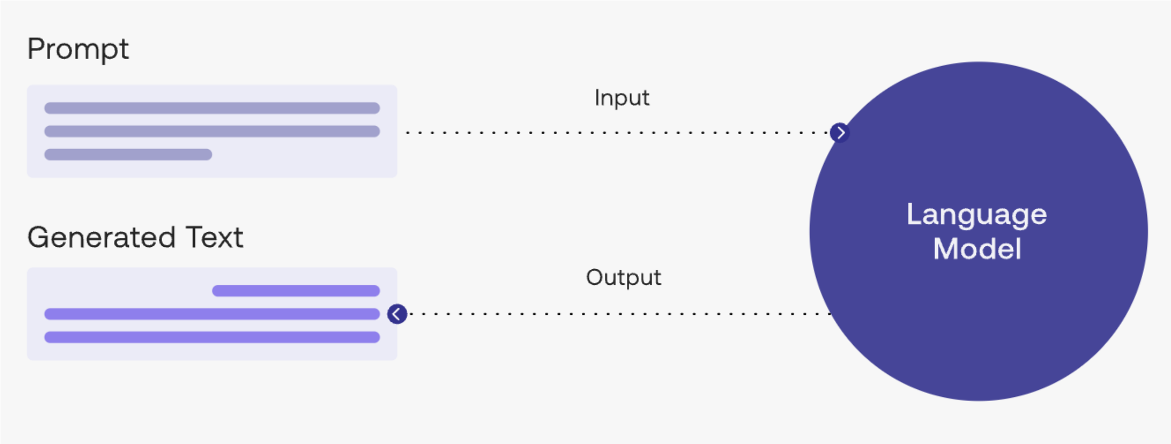
- AGI training cost is high > 1 million USD: big companies can cover?
- We have < 500 USD: how we can do?
- Foundation Models, Federated Learning, Transfer learning, Prompt Engineering
- Quantum computing: To accelerate AI training ➔ Quantum + AI is the future?

Foundation models towards AGI



國立中正大學 Cybersecurity Lab

Prompt engineering



- You must tell ChatGPT details as possible

Write a Python code to draw of bar chart from the following data Metric 2.4 GHz 5GHz 7GHz 10GHz 20GHz 30GHz 40GHz
Throughput 383 383 383 383 383 301192
Mean delay 3.5 3.5 3.5 3.5 3.5 69 152
Mean jitter 0.04 0.04 0.04 0.04 0.04 0.04 0.06

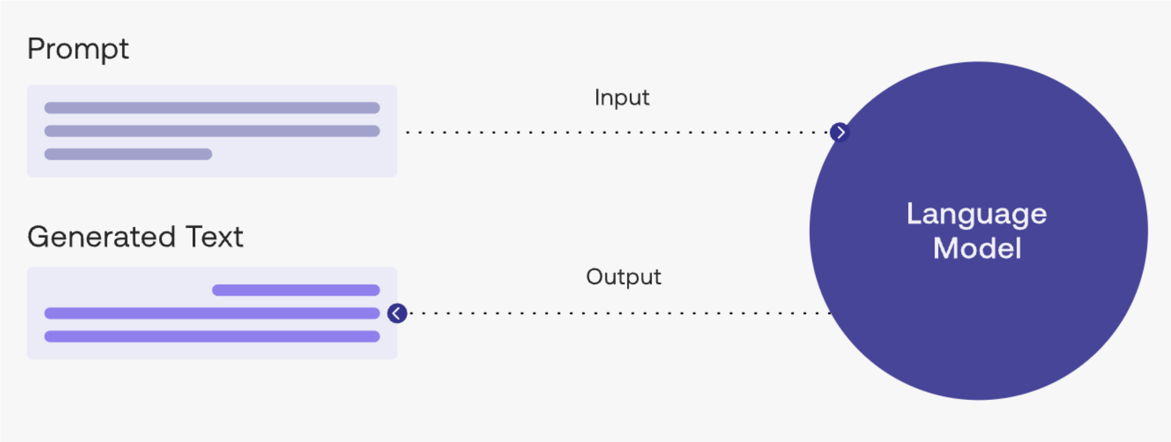
ChatGPT

Certainly! You can use the `matplotlib` library to create a bar chart from the provided data. Here's a simple Python script for that:

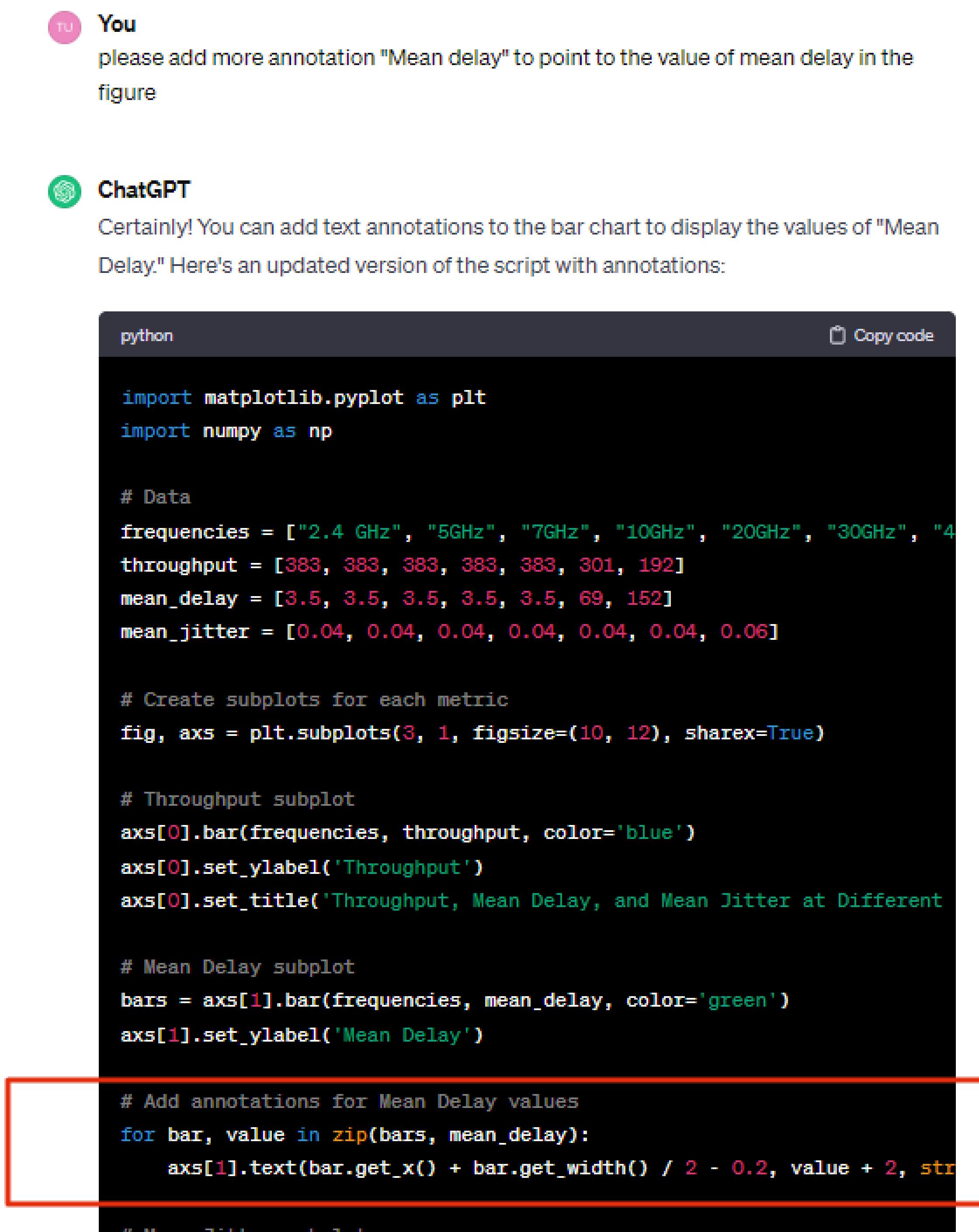
python Copy code

```python
import matplotlib.pyplot as plt
import numpy as np

# Data
frequencies = ["2.4 GHz", "5GHz", "7GHz", "10GHz", "20GHz", "30GHz", "4
throughput = [383, 383, 383, 383, 383, 301, 192]
mean_delay = [3.5, 3.5, 3.5, 3.5, 3.5, 69, 152]
mean_jitter = [0.04, 0.04, 0.04, 0.04, 0.04, 0.04, 0.06]

# Create subplots for each metric
fig, axs = plt.subplots(3, 1, figsize=(10, 12), sharex=True)

# Throughput subplot
axs[0].bar(frequencies, throughput, color='blue')
axs[0].set_ylabel('Throughput')
axs[0].set_title('Throughput, Mean Delay, and Mean Jitter at Different

# Mean Delay subplot
axs[1].bar(frequencies, mean_delay olor='green')
axs[1].set_ylabel('Mean Delay')
```

Prompt engineering



- You must tell ChatGPT details as possible



You
please add more annotation "Mean delay" to point to the value of mean delay in the figure

ChatGPT
Certainly! You can add text annotations to the bar chart to display the values of "Mean Delay." Here's an updated version of the script with annotations:

python Copy code

```python
import matplotlib.pyplot as plt
import numpy as np

# Data
frequencies = ["2.4 GHz", "5GHz", "7GHz", "10GHz", "20GHz", "30GHz", "4
throughput = [383, 383, 383, 383, 383, 301, 192]
mean_delay = [3.5, 3.5, 3.5, 3.5, 3.5, 69, 152]
mean_jitter = [0.04, 0.04, 0.04, 0.04, 0.04, 0.04, 0.06]

# Create subplots for each metric
fig, axs = plt.subplots(3, 1, figsize=(10, 12), sharex=True)

# Throughput subplot
axs[0].bar(frequencies, throughput, color='blue')
axs[0].set_ylabel('Throughput')
axs[0].set_title('Throughput, Mean Delay, and Mean Jitter at Different

# Mean Delay subplot
bars = axs[1].bar(frequencies, mean_delay, color='green')
axs[1].set_ylabel('Mean Delay')

# Add annotations for Mean Delay values
for bar, value in zip(bars, mean_delay):
    axs[1].text(bar.get_x() + bar.get_width() / 2 - 0.2, value + 2, str
```

Places to find State of the Art (SOTA)

Breadth of AI subjects

- www.PapersWithCode.com – for a given analysis type, compares papers on the same data set, link to code and data
  - Computer Vision (1,223 tasks, like these)
    - Image Classification – the image has a "cat" as the primary subject
    - Object Detection – a rectangle bounding box around the outside of each subject
    - Semantic Segmentation – a polygon around all examples of something, "people", "vehicles"
    - Instance Segmentation – a polygon around each instance of something, "person 1", "person 2", …
    - Image Generation – from text, generate images
  - Natural Language Processing (699 tasks)
    - Language modeling
    - Question answering
    - Machine translation
    - Text generation
    - Sentiment analysis
  - Medical
  - Time Series
  - Graphs
  - Speech
  - Audio
  - Computer Code
  - Reasoning
  - Playing Games
  - Robots
  - Adversarial
  - Knowledge Base

The Segment Anything Model (SAM), as a Foundation model, unifies most vision tasks in one architecture!

Src: Greg Makowsk

Places to find State of the Art (SOTA)

- Learn.DeepLearning.AI – Andrew Ng works with OpenAI and others to develop short courses
  - ChatGPT Prompt Engineering for Developers
  - LangChain for LLM Application Development
  - How Diffusion Models Work
  - Building Systems with the ChatGPT API
  - LangChain Chat with your Data
  - https://www.deeplearning.ai/courses/ and many others
- Podcast – This Week in Machine Learning (TWIML), at Episode 639 as of July 21, 2023
  - https://podcasts.google.com/search/twiml
  - https://twimlai.com/
  - 30 – 60 minute interviews with applied researchers, leaders
  - By Sam Charrington
  - sponsored
- YouTube.com channels covering LLMs and AI, in no particular order (found by searching for specific topics, and I returned).
  - Lex Fridman: 1-3 hour interviews with excellent people
  - EYE ON AI, Craig Smith: good interviews
  - TheAIGRID
  - Yannic Kilcher: goes over papers and software
  - Peter H Diamandis
  - RoboFlow: papers, hilghlights
  - David Shapiro ~ AI
  - Edan Meyer
  - SFbayACM: 100+ AI talks like this one
  - Anastasia Marchenkova: a Quantum Computing researcher
  - Future of Life Institute: Future of AI is one of 4 focus areas
  - Maziar Raissi: Many AI class lectures
- Blogs, Hubs
  - www.TowardsDataScience.com
  - www.KDNuggets.com - one of the oldest DS/AI Hubs

Src: Greg Makowski

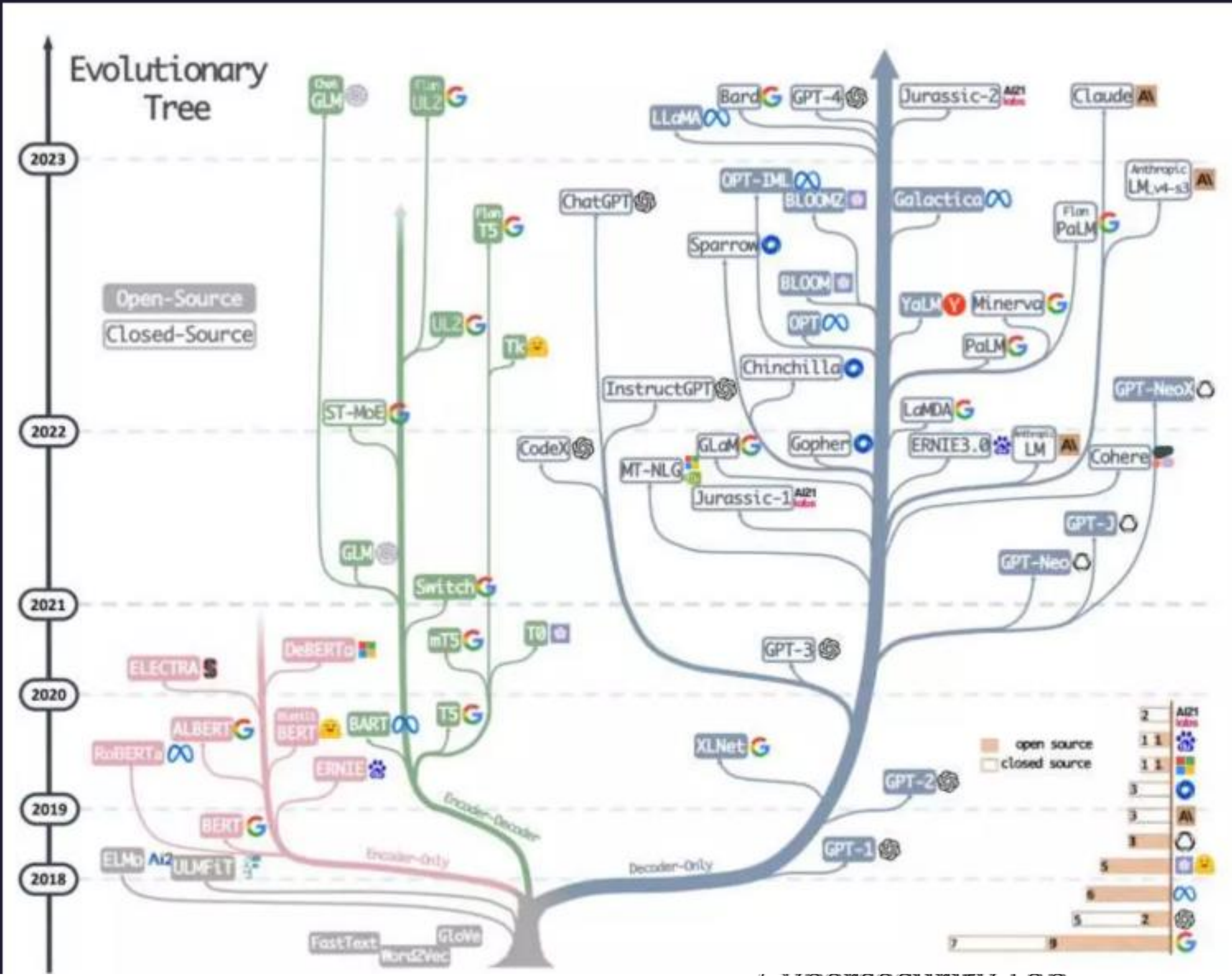
There is a lot of LLM Evolution in a short time

- Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond
- April 2023
- https://arxiv.org/pdf/2304.13712.pdf

- NOTE: "open source" does not always mean "available for commercial use"

- LLaMA 2 came out July 18, 2023, and is available for commercial use

Src: Greg Makowski



Summary

- Technology progress is rapid

- If you are working in this field, you may often update the latest ones via 3GPP, ETSI, ITU, Qualcomm/Intel/Huawei

Final exam

- Place: Room 326/ Room 339 (two rooms)
- Time: 1/4 13:20~15:20 (2 Hours)

- The question format as the mid-term exam