



**STAC-AI™**

*From the model to the metal™*

Jack Gidding  
CEO, STAC

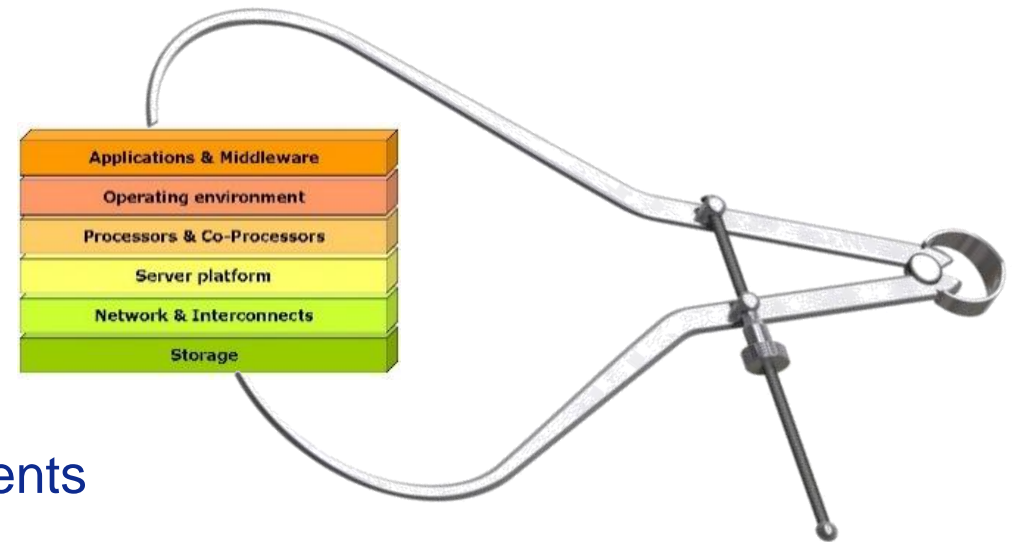
[jack.gidding@STACresearch.com](mailto:jack.gidding@STACresearch.com)

# Motivations for STAC-AI

GenAI has become a strategic technical challenge in finance

Council members have asked us to facilitate benchmarks that are:

- Finance relevant (understandable by the business)
  - Corpora
  - Queries
  - Workload patterns
  - Measurements
- Useful for engineers
  - Full stack
  - Implementation agnostic
  - Additional internal (non-benchmark) measurements
  - Meticulously documented
- Adjudicated by a trusted, independent 3<sup>rd</sup> party



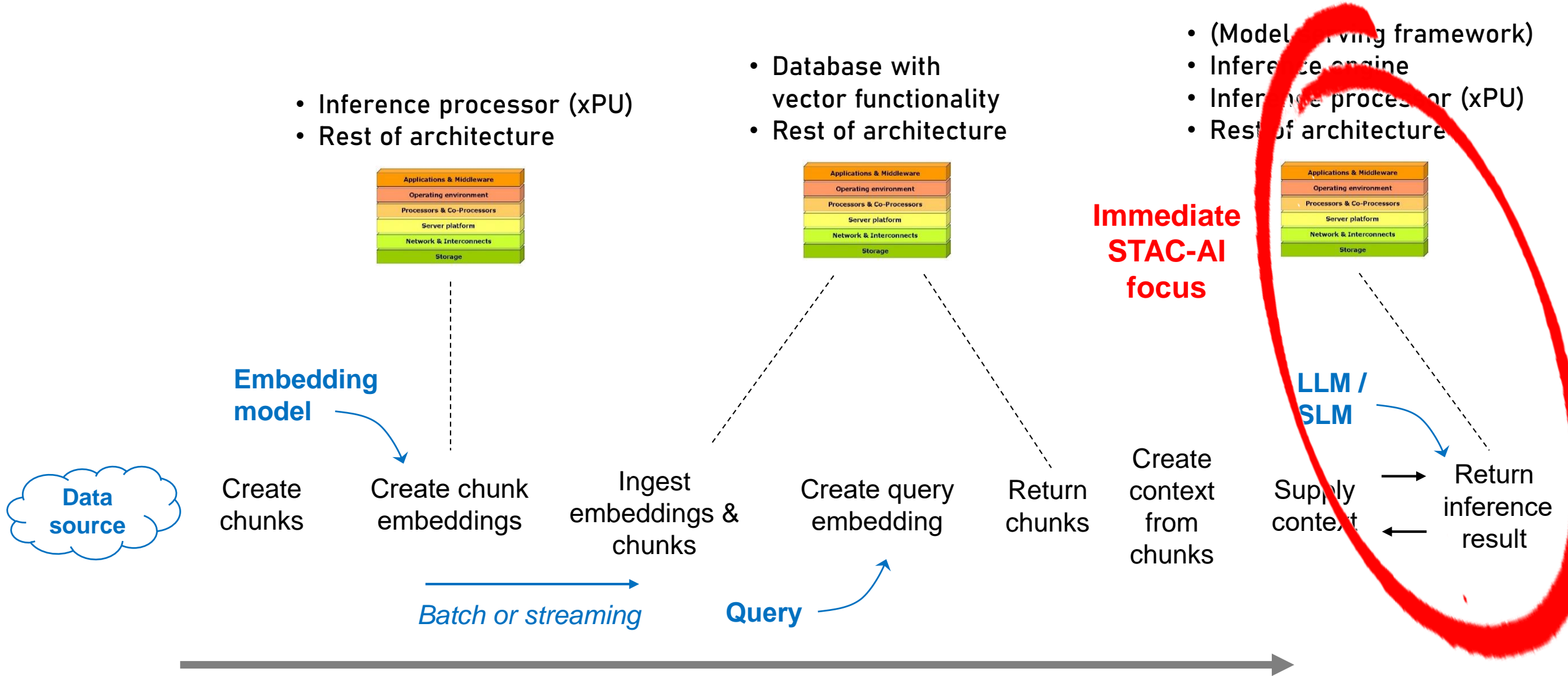
# STAC-AI Working Group

- 10 financial firms (more welcome!)
  - Banks, hedge funds, exchanges
- 15 vendor firms
  - NVIDIA
  - AMD
  - Intel
  - Groq
  - Myrtle
  - NeuReality
  - AWS
  - Microsoft
  - Oracle
  - IBM
  - HPE
  - Dell
  - KX
  - Weka
  - Pure
- Many potential workflows in scope
  - Training, fine-tuning, agentic workflows
- Initial target:
  - Retrieval-augmented generation (RAG)

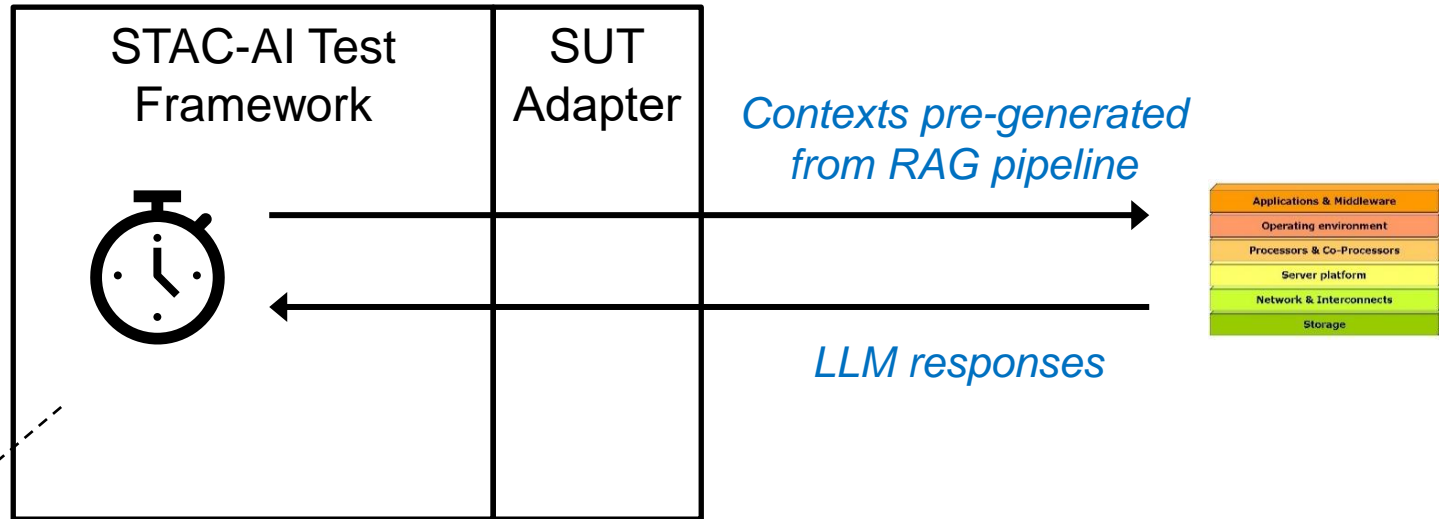
# RAG (retrieval augmented generation)

- Rough broad version of the definition:
  - Using data retrieved from someplace to augment what you send to an LLM
  - Unstructured content, RDBMS, graph DB, time series database
- Purpose is to make LLM responses more:
  - Relevant (including timely)
  - Accurate (minimize hallucinations)
- In classic RAG:
  - The data source is unstructured
  - Retrieval uses similarity search – Find me the top K chunks of content whose meaning is most similar to the meaning of the query
  - Similarity search does distance calculations between a vector (an “embedding”) corresponding to the query and embeddings corresponding to the content chunks
  - Content embeddings are created beforehand by a special model separate from the LLM

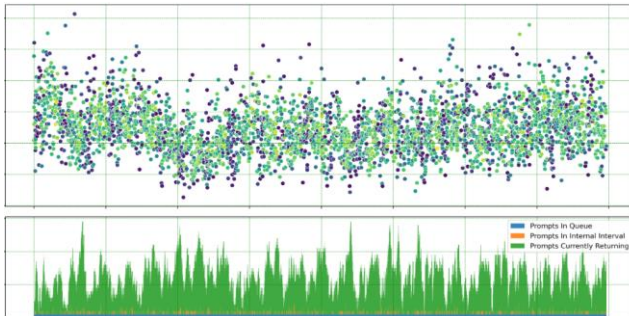
# High-level RAG workflow (classical)



# RAG-based tests isolating the inference step



Analysis



# STAC-AI Status

- Test harness framework implemented
- STAC and vendors contributing additional code
- Preliminary use cases in place
- Several working group members trying it out
- So is STAC (more on this in a bit)
- Collecting feedback
- Final specs and harness this summer

# Measurements

- Reaction time
  - Time to first characters
- Response time
  - Time to complete response
- Smoothness (for chat use cases)
  - More on this in a sec
- Efficiency
  - Responses per dollar / energy / space
- Quality
  - “Goodness” (output of Model X on **SUT** vs some **ideal output**) – not Day 1
  - Fidelity (output of Model X on **SUT** vs output of Model X on **reference SUT**) – Day 1



# Smoothness vs Response Time

## Prompt: What are some unusual ideas for home-made birthday cakes?

Oooh, let's get creative with those cakes! Here are a few unusual ideas for home-made birthday cakes:

- Meringue-Topped, Layered Ice Cream Cake: Imagine layers of ice cream, frosting, and a meringue topping. Yum!
- Vanilla Cake with Vanilla Buttercream Frosting: A classic cake with a twist—try adding colorful sprinkles or edible glitter for a sparkly surprise.
- Homemade Confetti Cake: Make a vanilla or chocolate cake, then mix in colorful sprinkles or chocolate chips for a fun and festive look.
- Berry Chantilly Cake: Layers of white cake, fresh berries, and Chantilly cream make this a light and refreshing option.

**Better smoothness, worse response time**

Oooh, let's get creative with those cakes! Here are a few unusual ideas for home-made birthday cakes:

Meringue-Topped, Layered Ice Cream Cake: Imagine layers of ice cream, frosting, and a meringue topping. Yum!

Vanilla Cake with Vanilla Buttercream Frosting: A classic cake with a twist—try adding colorful sprinkles or edible glitter for a sparkly surprise.

Homemade Confetti Cake: Make a vanilla or chocolate cake, then mix in colorful sprinkles or chocolate chips for a fun and festive look.

Berry Chantilly Cake: Layers of white cake, fresh berries, and Chantilly cream make this a light and refreshing option.

**Better response time, worse smoothness**

# STAC-AI for both self-hosted and 3<sup>rd</sup> party LLMs

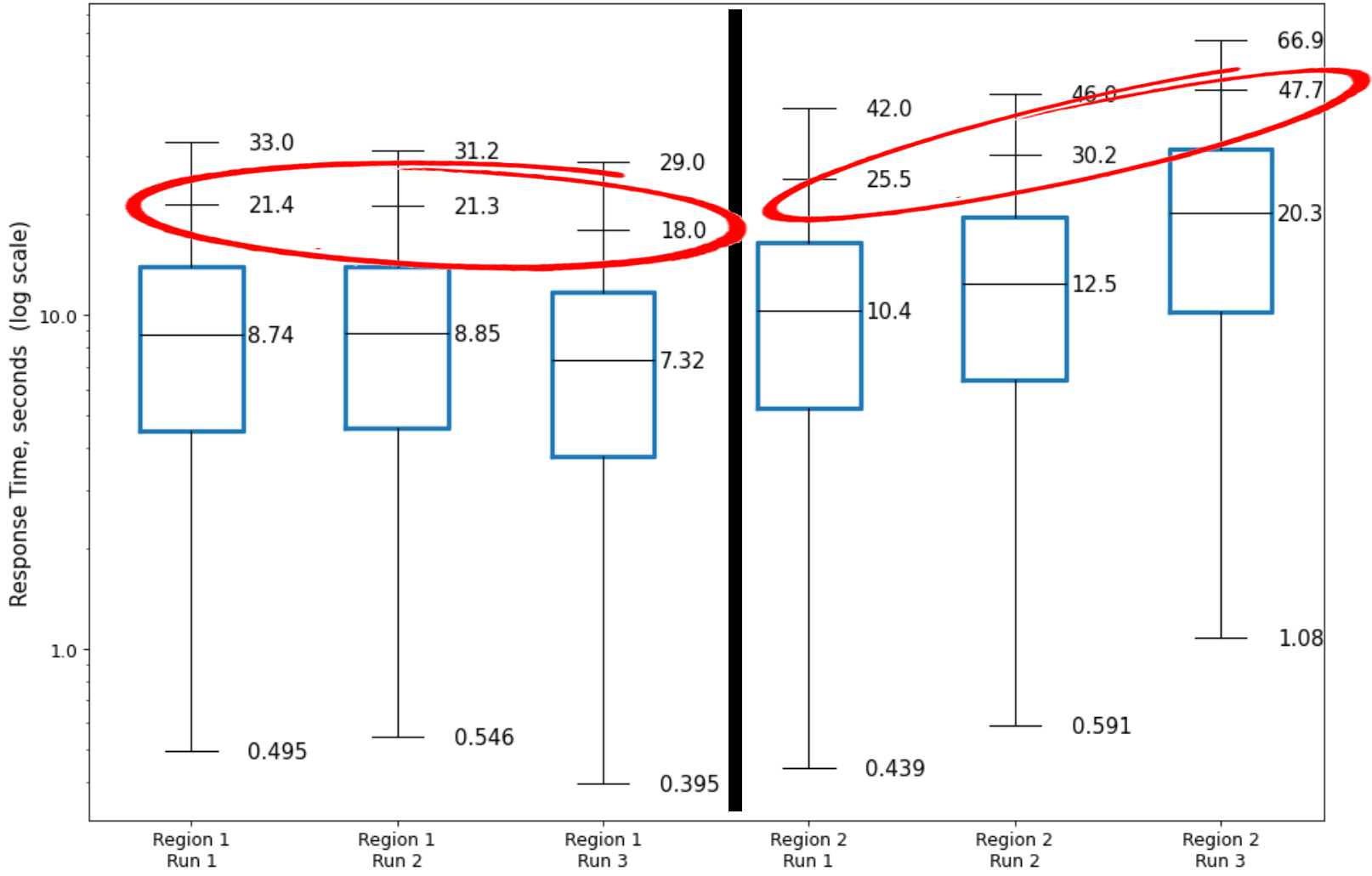
- STAC-AI can test self-hosted models. E.g.:
  - Mistral 7B, model serving framework W, instance type X, region Y, cloud provider Z
  - DBRX, inference engine X, on prem on GPU server Y
- But it is just as valid with 3<sup>rd</sup>-party LLMs. E.g.:
  - [GPT 4, Claude 3, Gemini 1.5, Command R+, SenseNova 5] in region X on cloud provider Y with provisioned throughput N

# Early test case using STAC-AI

- A large bank asked STAC to help evaluate 3<sup>rd</sup>-party LLM endpoints at a CSP
  - (Multiple models) × (multiple regions)
- Question:
  - Suppose a data scientist has chosen Model A for a given task
  - How does Model A perform in each geographic region?
- Info will inform solution architecture and capacity planning
- Use case:
  - EDGAR filings (10-K) for Russell 3000 stocks over the last 5 years
  - Summarization task that requires comparisons across companies
  - Contexts were limited to fit in the models' context windows
- Work is ongoing

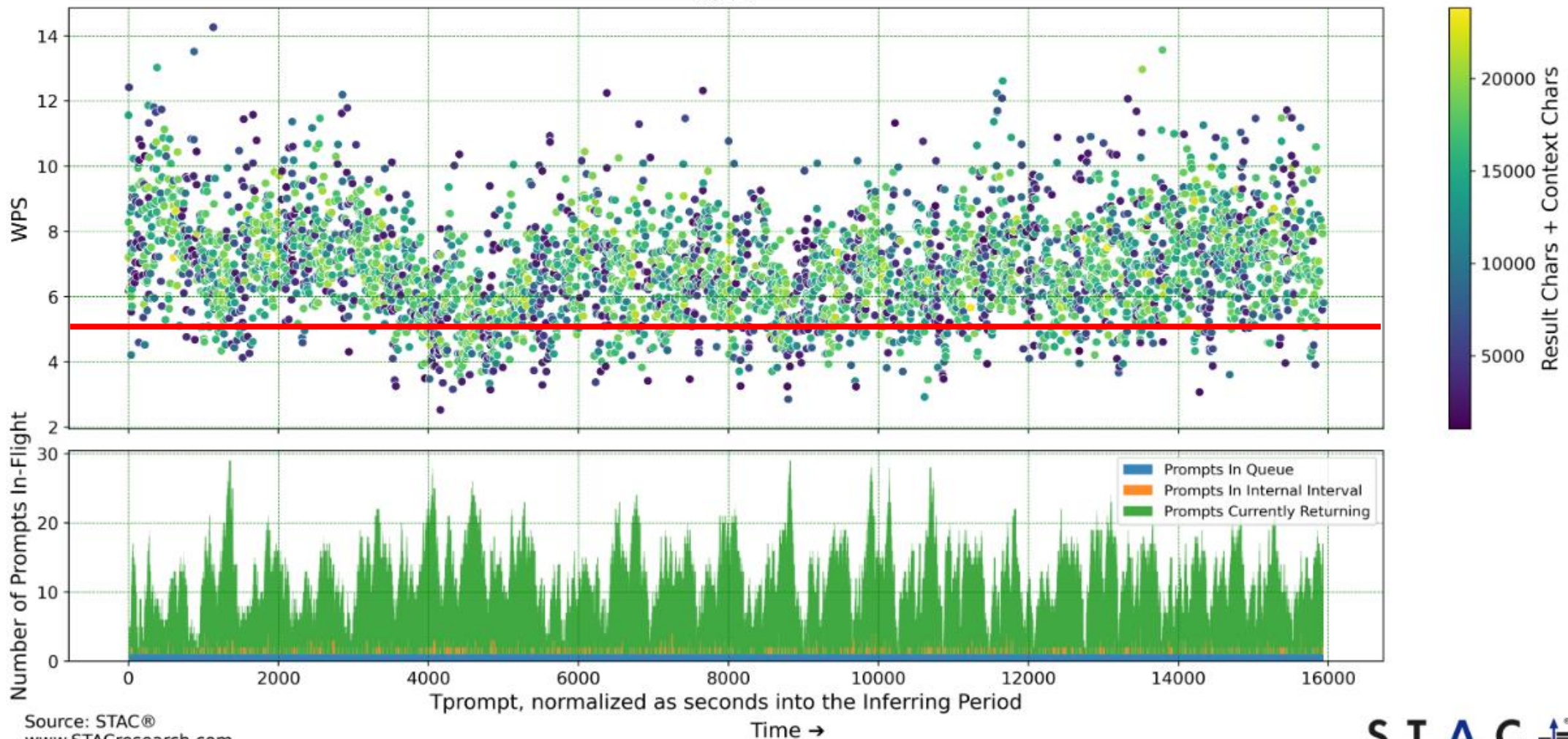
# Example comparison – Response Times, Model 1, two regions

Response Times for Model 1 across Regions and Runs  
Minimum, Median, 95P and Maximum Annotated



# Example time analysis: Model 2 in Region 2

Words per Second (WPS, excluding prefill) over Time  
Run 1



Source: STAC®  
www.STACresearch.com  
Copyright © 2024 STAC

———— = top of typical human reading speed range (English)



# Ways to get involved

- Join the working group!
- Talk to us if STAC-AI might help you with a project

[www.STACresearch.com/ai](http://www.STACresearch.com/ai)