Talifun Tokenizer

Lowest Latency
Highest Throughput

Get measurable performance gains across your AI workloads

Built for high-throughput AI pipelines

Talifun Tokenizer removes tokenization as a bottleneck across production AI workloads.

FAST

- Sub-millisecond p99 latency
- GB/s throughput
- Scales across parallel execution

DROP-IN

- Use existing hardware
- Keep your software architecture
- Keep token IDs and counts exactly the same

CONVENIENT

- Source code access
- Python, Node.js, and Rust packages
- Unlimited lifetime use license
- Low risk, easy to measure results

Throughput and p99 latency

Talifun vs next fastest tokenizer - o200k model

See all benchmarks

	THROUGHPUT	THROUGHPUT UPLIFT	P99 LATENCY	P99 REDUCTION
Python	**832.65 MB/s**	**19× faster**	**0.34 ms**	**6.9× lower**
Node.js	**928.39 MB/s**	**9.5× faster**	**0.40 ms**	**6.8× lower**
Rust	**943.20 MB/s**	**9.5× faster**	**0.23 ms**	**5.6× lower**

Modelled saving across model sizes

Pilot will capture your actual workload savings.

Use case calculator

	1B model 64 × 1M token pool		70B model 64 × 1M token pool		405B model 64 × 1M token pool	
	REQUEST LATENCY SAVED	POOL TOKENS SERVED	REQUEST LATENCY SAVED	POOL TOKENS SERVED	REQUEST LATENCY SAVED	POOL TOKENS SERVED
Inference / Chat Pipeline	9.20%–34.5%	10.1%–52.8%	4.27%–14.2%	4.46%–16.5%	1.41%–4.42%	1.43%–4.62%
API Gateway Token Accounting	12.7%–76.0%	14.6%–315.9%	12.7%–76.0%	14.6%–315.9%	12.7%–76.0%	14.6%–315.9%
Online RAG Query-Time	12.2%–18.5%	13.9%–22.8%	7.90%–11.8%	8.58%–13.3%	3.39%–4.97%	3.51%–5.23%
RAG Ingest / Indexing	3.51%–5.45%	3.64%–5.76%	3.51%–5.45%	3.64%–5.76%	3.51%–5.45%	3.64%–5.76%
Embedding / Reranking	4.11%–25.7%	4.29%–34.6%	0.74%–9.02%	0.74%–9.91%	0.15%–2.19%	0.15%–2.24%
Evaluation / Regression	30.6%–37.2%	44.1%–59.1%	30.6%–37.2%	44.1%–59.1%	30.6%–37.2%	44.1%–59.1%

Your workloads are waiting on Tokenization

Tokenization is the step that turns human text into the numbered pieces an AI model understands. It sits directly in the critical path to your AI workloads.

SERVING

- Prompt preparation
- Padding choices
- Scheduling efficiency
- Routing
- Batch admission

RAG

- Chunking
- Query assembly
- Retrieved context
- Reranking
- Repeated token counting
- Token materialization

EMBEDDINGS

- Batch creation
- Input shaping
- Tokenizer-paced throughput
- Pre-vectorization bottleneck

EVALS

- Large regression suites
- Safety suites
- Prompt re-tokenization
- Reference re-tokenization
- Trace re-tokenization
- Generated output re-tokenization

GATEWAYS

- Token accounting
- Metering
- Rate limits
- Routing
- Every request path

Context windows keep getting larger

Tokenization performance costs are becoming visible in production metrics as prompts carry more of the workload into every model call.

Classic Request

Short prompt, one model call

User prompt

Context window

short user prompt

Token size 1k-32k tokens

Modern Request

Context assembled from many sources

Instructions
User Prompt
Retrieved documents
Tool outputs
Code
Logs & Records

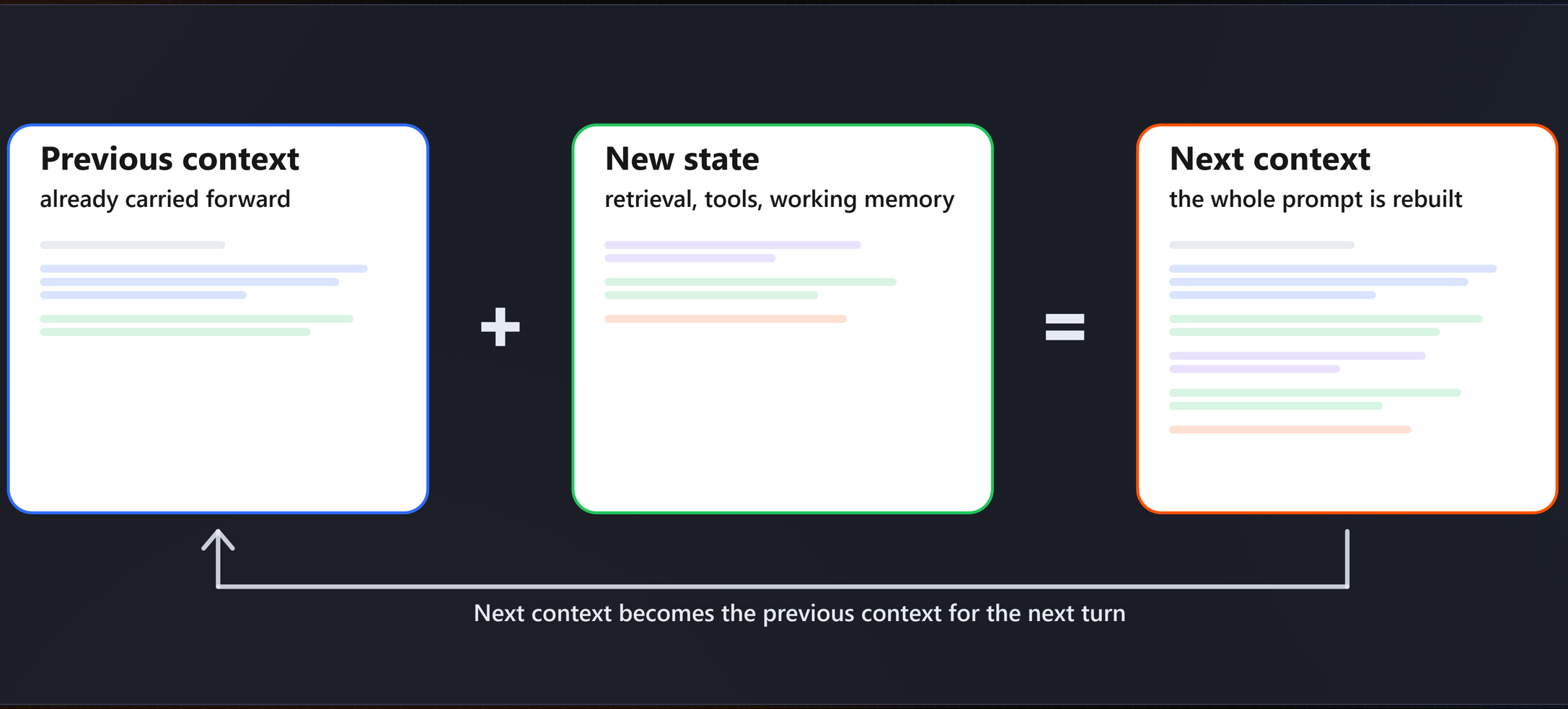
Context window

long prompt / larger input

Token size 128k-1m tokens

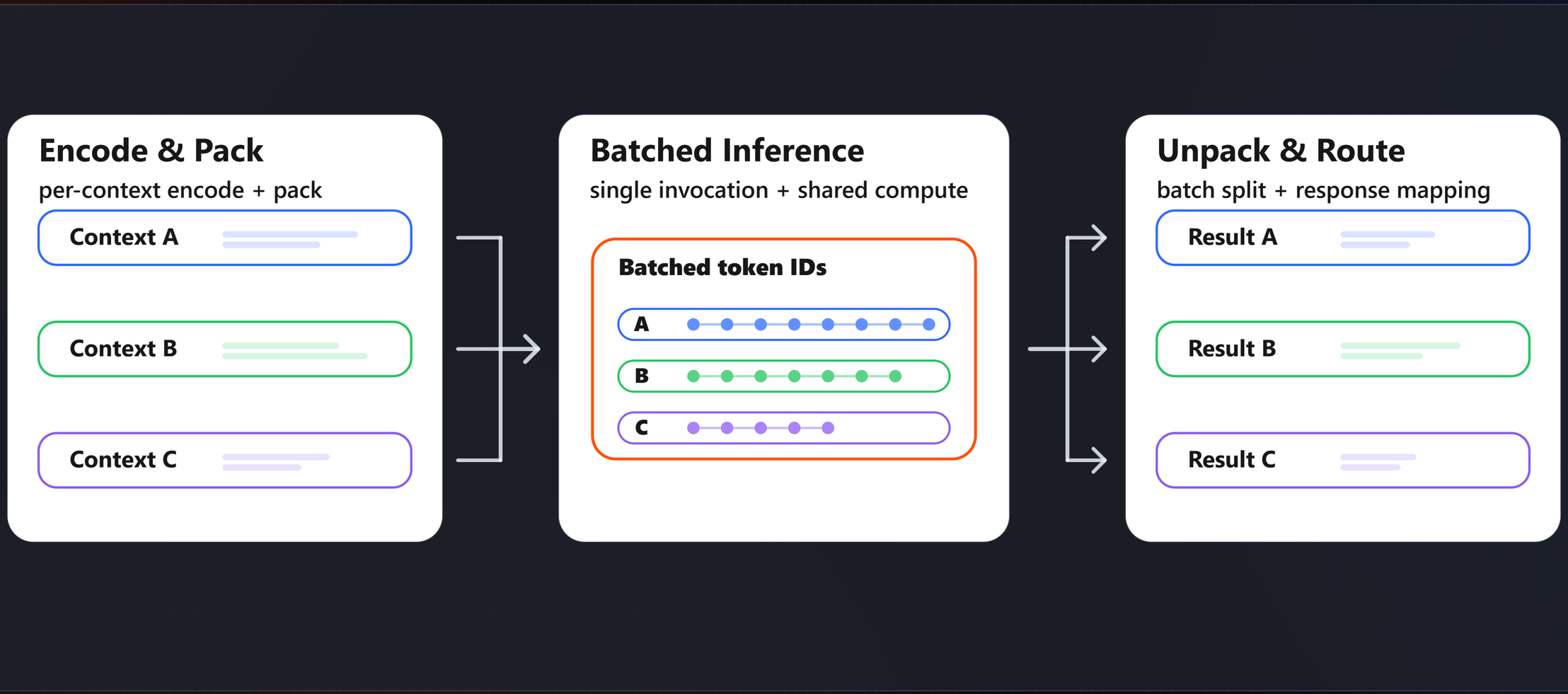
Each exchange cumulatively grows the context

Each reply carries the conversation so far, adds new text, and sends a larger prompt through tokenization again.



Fill the inference buffer for maximum throughput

Batch multiple similar sized contexts with as little padding as possible.



What Talifun Tokenizer can do for your workloads

Faster tokenization turns into faster experiences, more capacity, and lower operational overhead.

LATENCY

Faster responses

Make customer-facing AI feel quicker, with less waiting before an answer starts.

CAPACITY

More requests

Serve more users and traffic spikes without immediately expanding your infrastructure.

FRESHNESS

Faster RAG ingest

Bring new documents and updates into search experiences sooner.

UTILIZATION

Better batch efficiency

Get more value from the hardware you already pay for during busy periods.

OVERHEAD

Reduced gateway overhead

Keep customer traffic moving smoothly while still managing usage and access.

Talifun fits everywhere text is converted to tokens

From chat and RAG to gateways, embeddings, training, and evals, Talifun accelerates the repeated conversion work every pipeline depends on.

Inference / Chat Pipeline

- Request intake
- Build prompt
- **Tokenize**
- Run prefill/decode
- Write response

Online RAG Query-Time

- Query intake
- Retrieve context
- Rerank/select
- Build prompt
- **Tokenize**
- Call model
- Write answer

Agentic RAG Orchestration

- Task intake
- Run planner
- Call tools/retrieval
- Rebuild state
- **Tokenize**
- Run reasoning step
- Write final response

API Gateway Token Accounting

- Request intake
- Evaluate policy
- **Tokenize**
- Decide admission
- Update quota/billing
- Route request

Moderation / Classification

- Message intake
- **Tokenize**
- Run classifier
- Decide policy
- Gate downstream

Embedding / Reranking

- Load items
- **Tokenize**
- Pack batch
- Run embed/rerank
- Write vectors/scores

RAG Ingest / Indexing

- Load documents
- Parse/extract
- Chunk text
- **Tokenize**
- Embed chunks
- Write index

Offline Training Corpus Build

- Load corpus
- Filter/dedupe
- **Tokenize**
- Pack sequences
- Write shards
- Finalize dataset

Online Training Input Pipeline

- Loader read
- **Tokenize**
- Prefetch
- H2D transfer
- Run GPU step
- Handoff batch

Evaluation / Regression

- Load prompt suite
- Expand variants
- **Tokenize**
- Run evals
- Compute metrics
- Decide release

Founder-led partnership, with industrial-grade execution

Customers get direct access to the builder, fast technical judgment, and accountable ownership on this specialist topic.

FOUNDER-LED SUPPORT

- Direct access to the builder
- Issues
- Traces
- Correctness questions
- Rollout decisions

STABILITY & SYSTEMS

- Documented integration path
- Correctness testing
- Pinned packages
- Rollback planning
- Trace-backed acceptance criteria

TRACTION-LED PROOF

- Buying case based on measured impact
- Real traces
- Benchmark deltas
- CPU profile
- Rollout risk

Run a paid focused pilot

We bring the tokenizer, benchmark support, and, where permitted, hands-on help making the integration changes. You bring representative workloads, the target model configuration, and the current tokenizer behavior to compare against.

OFFER

Trace-backed paid pilot

Use real prompts, RAG paths, gateway accounting, batch jobs, or eval workloads to prove where faster tokenization changes the customer outcome. If permitted, we help make the integration changes.

WHAT WE NEED

A measured path to replay

Representative traces, model configuration, expected token IDs and counts test cases, and the latency, throughput, or CPU metrics that matter.

DECISION

Evidence-backed recommendation

Measured impact, correctness evidence, and a clear go/no-go decision for production adoption.

01 Configure tokenizer for model

02 Replace tokenizer call

03 Verify outputs

04 Review metrics

05 Decision