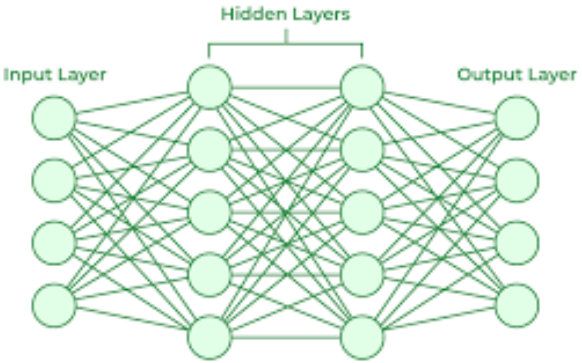
Small Language Models

Michael Mollel, PhD
Sartify Company Limited

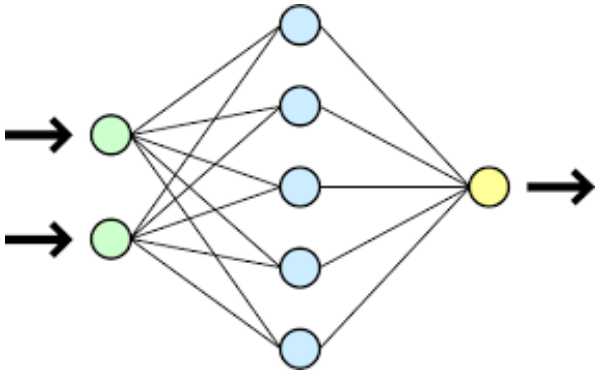
Small Language Model (SLM)

SML - Compact, highly efficient versions of massive large language models



Key Characteristics:

- Fewer Parameters: Typically, under 10 billion (vs. Hundreds of billions in LLMs)
- Resource Efficient: Lower computational costs and energy usage
- Task-Focused: Trained on smaller, specialised datasets
- Balanced Performance: Maintains efficiency without sacrificing too much capability



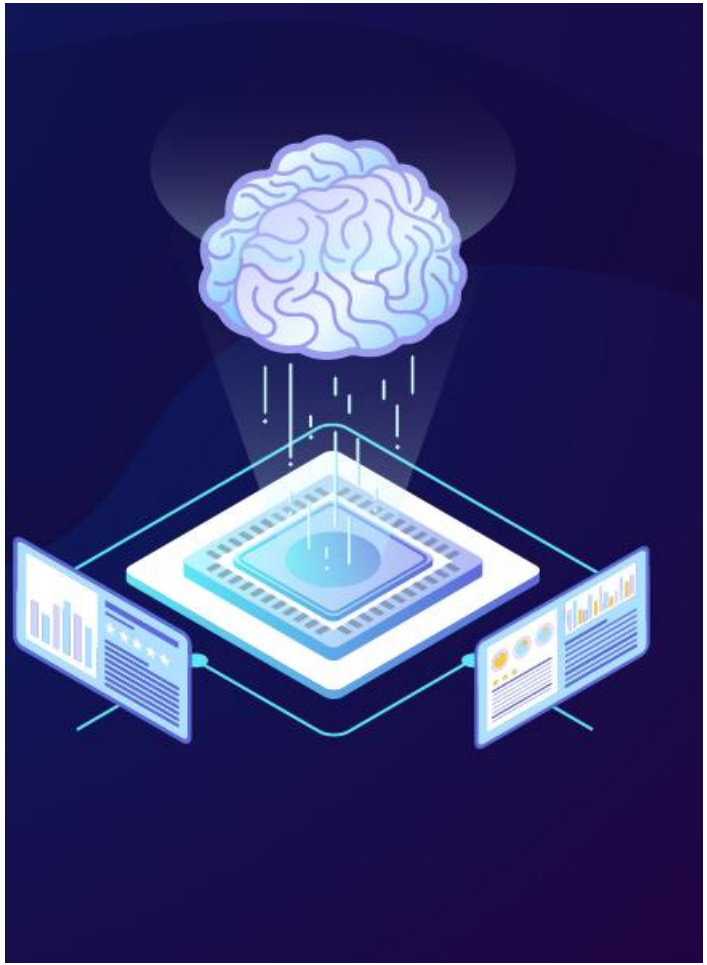
Why SLMs Matter

The Problem: LLMs are resource-intensive and inaccessible to many

The Solution: SLMs provide:

⚡ **Efficiency**: Run on limited computational power

- Don't need massive computational power
- Perfect for smartphones, tablets, IoT devices



Why SLMs Matter

The Problem: LLMs are resource-intensive and inaccessible to many

The Solution: SLMs provide:

💡 **Accessibility**: Affordable for smaller budgets

- Suitable for on-premise deployments

🏠 • Enhanced privacy and data security

- No constant cloud dependency

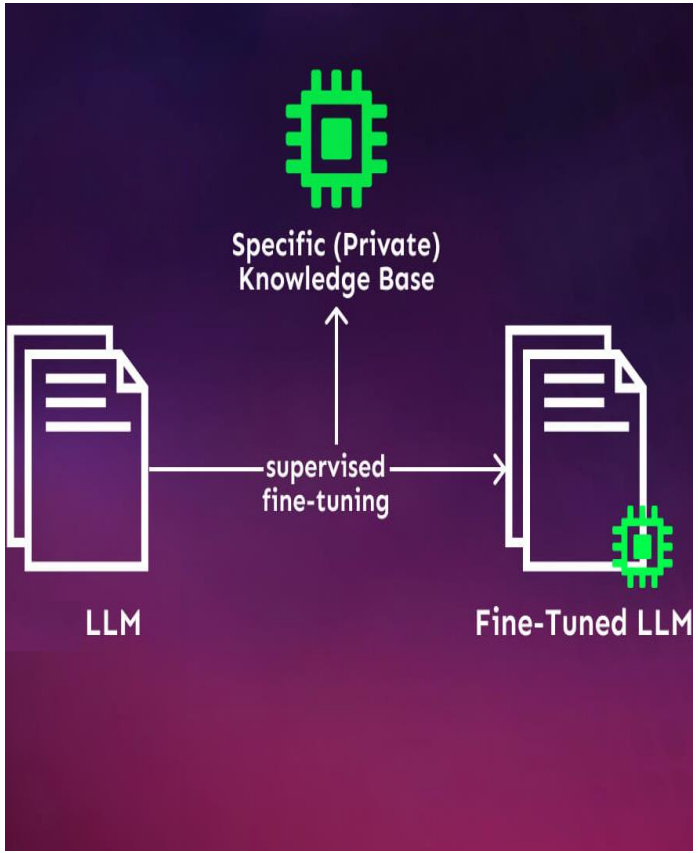
Why SLMs Matter

The Problem: LLMs are resource-intensive and inaccessible to many

The Solution: SLMs provide:

🎯 **Customisation:** Easy to fine-tune for specific tasks

- Quick adaptation to niche tasks
- Specialised domains (healthcare, education, customer support)



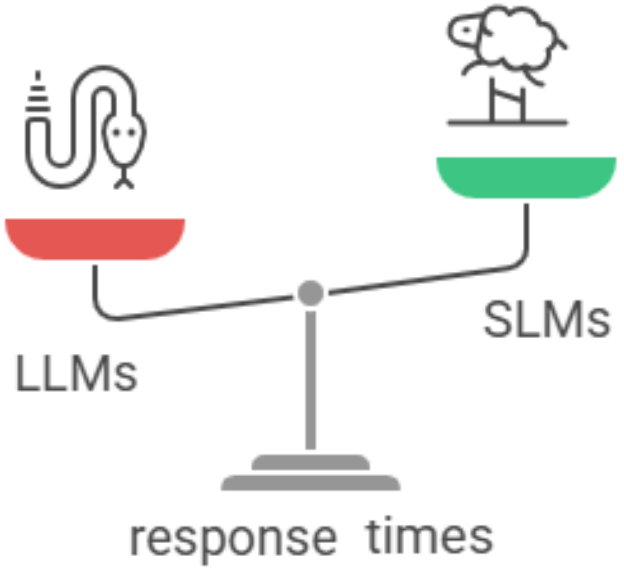
Why SLMs Matter

The Problem: LLMs are resource-intensive and inaccessible to many

The Solution: SLMs provide:

🚀 **Speed**: Faster inference and response times

- Faster response times
- Perfect for real-time applications



How SLMs Work

Next Word Prediction

- Analyse patterns from training text
- Predict the most likely next word in the sequence.
- Example: ”The names are as follow Michael ..." → ”Juma”

Transformer Architecture

- Self-attention mechanism
- Understands word relationships and context
- Distinguishes meaning based on context

Size-Performance Balance

- Fewer parameters = less computational power
- Faster processing for real-time applications
- Specialized performance in focused domains

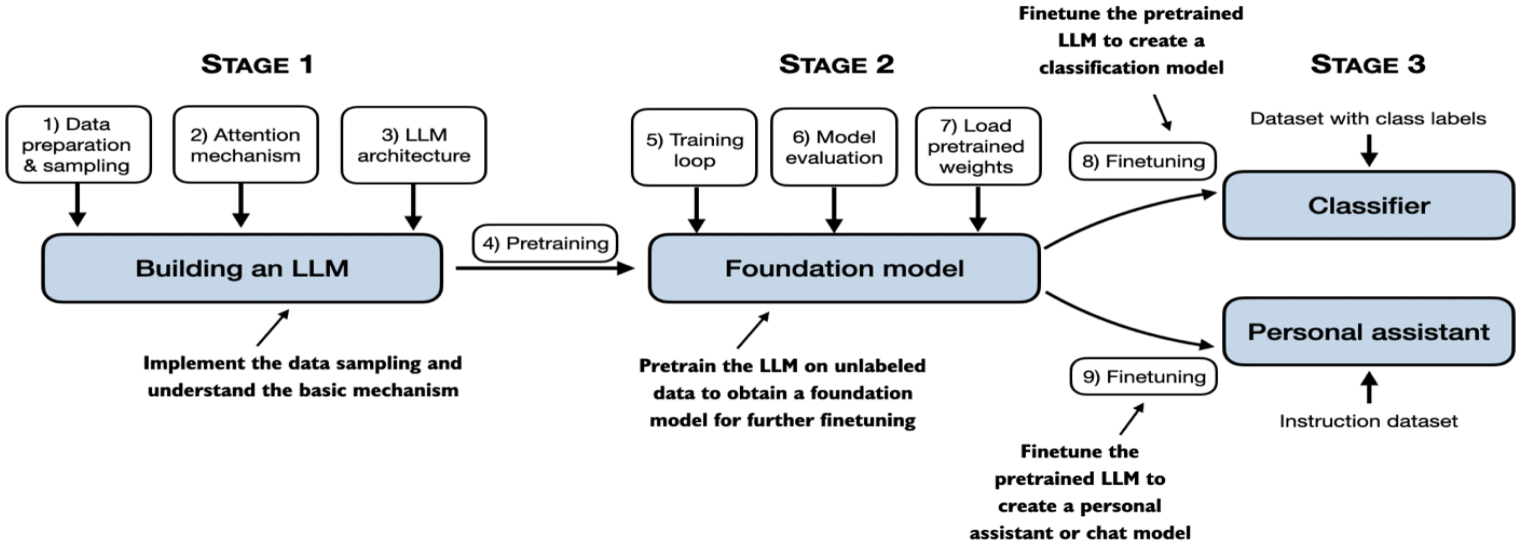
SLM Examples

MODEL	Parameters	Features
Pawa-Min	2B (**open**)	Swahili, Scalable
Llama 3.1	8B	Balanced power & efficiency
TinyLlama	1.1B	Mobile & edge optimized

SLM Creation Techniques

🚀Training from Scratch

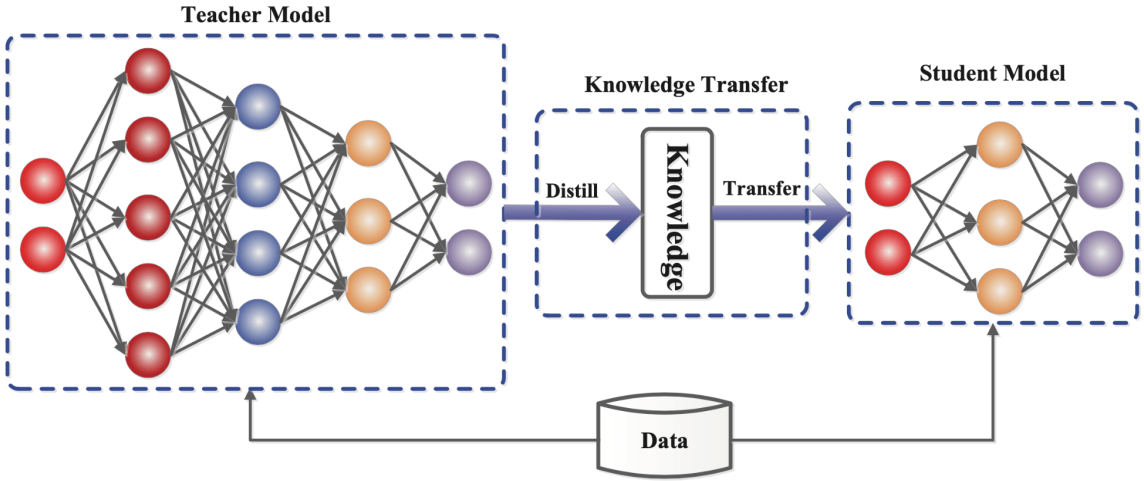
- Design compact architectures from the ground up
- Focus on efficient operations (depthwise convolutions, attention mechanisms)
- Optimise for target hardware constraints and use cases
- Requires extensive hyperparameter tuning and data preparation



SLM Creation Techniques

Knowledge Distillation

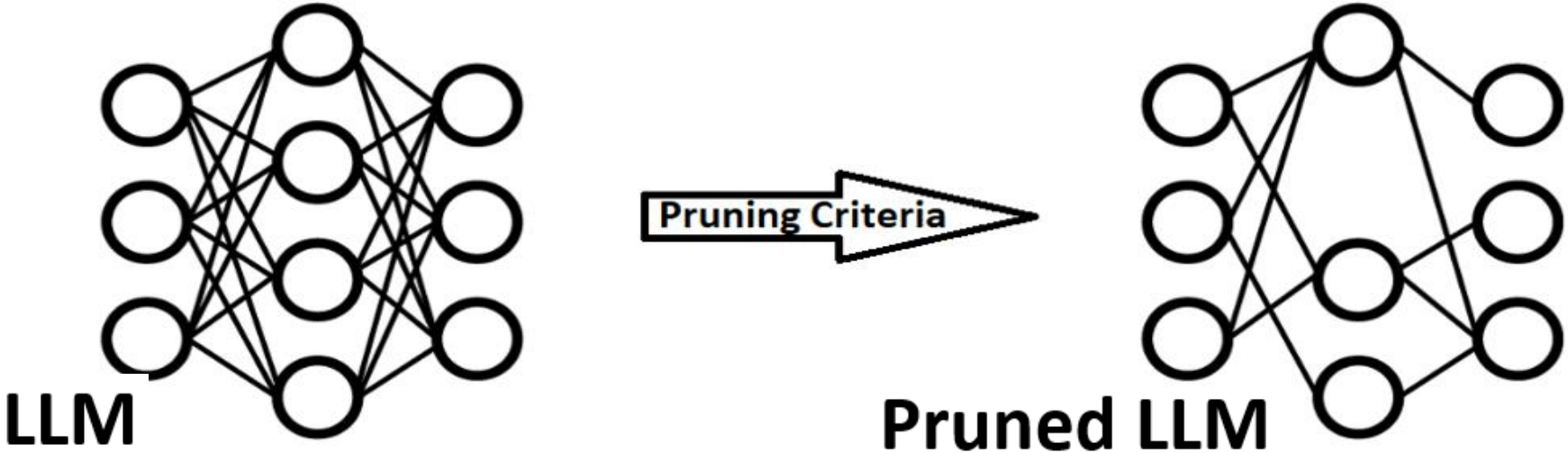
- Transfer knowledge from a large "teacher" to a small "student" model
- Methods: Response-based, Feature-based, Relation-based
- Retains accuracy while reducing size



SLM Creation Techniques

✂ Pruning

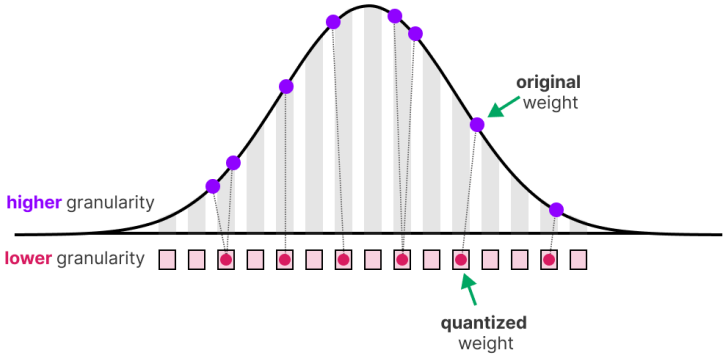
- Remove less essential neurons/parameters
- Trim unnecessary components
- Maintain performance while reducing size



SLM Creation Techniques

Quantization

- Use fewer bits to store numbers (32-bit → 8-bit)
- Reduce memory usage and increase speed
- Minimal impact on accuracy



Weights
(32-bit float)

2.52	-1.12	1.74	0.05
0.08	-0.22	-1.21	2.65
-0.13	1.60	0.02	-1.31
2.13	-0.01	1.83	1.65

Quantization

Quantized Weights
(8-bit signed int)

121	-54	83	2
4	-11	-58	127
-6	77	1	-63
102	0	88	79

Dequantization

Reconstructed Weights
(32-bit float)

2.53	-1.13	1.73	0.04
0.08	-0.23	-1.21	2.65
-0.13	1.61	0.02	-1.32
2.12	0.00	1.84	1.65

LLMs vs SLMs - Task Complexity

🎯 LLMs Excel At:

- Complex, sophisticated, general tasks
- Deep understanding and reasoning
- Long content creation
- Better accuracy across diverse tasks
- Long-range context understanding

⚡ SLMs Excel At:

- Simpler, focused tasks
- Specialized applications
- Domain-specific expertise
- Quick, efficient responses
- Resource-constrained environments

LLMs vs SLMs - Resource Constraints

LLMs Requirements:

- Significant computational power and memory
- Specialized hardware (GPUs)
- Higher operational costs
- Longer training times

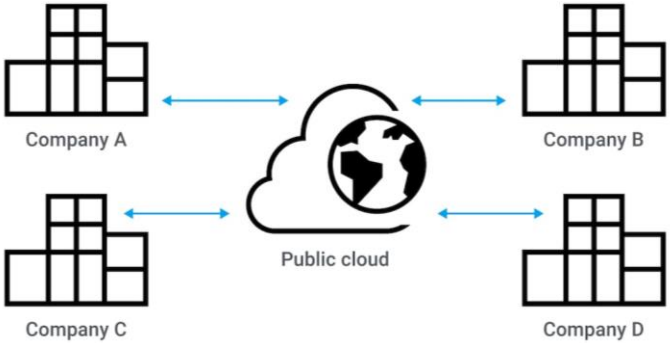
SLMs Advantages:

- Economical resource consumption
- Run on standard hardware
- Can operate on Raspberry Pi or smartphones
- Shorter training times
- Quick deployment capability

LLMs vs SLMs - Deployment Environment

LLMs Best For:

- Cloud environments with abundant resources
- High accuracy requirements
- Complex reasoning tasks
- Continuous internet connectivity



SLMs Best For:

- On-device AI applications
- Edge computing scenarios
- Offline functionality requirements
- Low-latency applications
- Privacy-sensitive environments

Choosing Between LLMs and SLMs

Task Complexity

How sophisticated are your requirements? **Complex reasoning** and multi-step tasks favour LLMs, while **focused, specific tasks** work well with SLMs.

Resource Availability

What's your computational budget? Consider **GPU memory, processing power**, and ongoing operational costs for your deployment scenario.

Deployment Location

Cloud deployment enables powerful LLMs, while **edge and on-device** scenarios typically require SLMs for practical performance.

Internet Connectivity

Always online or offline capability needed? **Intermittent connectivity** scenarios favour local SLMs over cloud-dependent LLMs.

Privacy Concerns

Can data leave the device? **Sensitive data** processing often requires local SLMs, while cloud-based LLMs suit less sensitive applications.

Latency Requirements

How fast do you need responses? **Real-time applications** often need SLMs, while **batch processing** can accommodate slower LLMs.

Test SLM Pawa Open Model 2B

Sartify HF:

- https://huggingface.co/sartifyllc/pawa-min-alpha

Sartify GitHub:

- https://github.com/Sartify/IndabaX_SLM

Google Colab:

- **Run:** SLM_Pawa.ipynb

SLM

🎯 SLMs Make AI Accessible

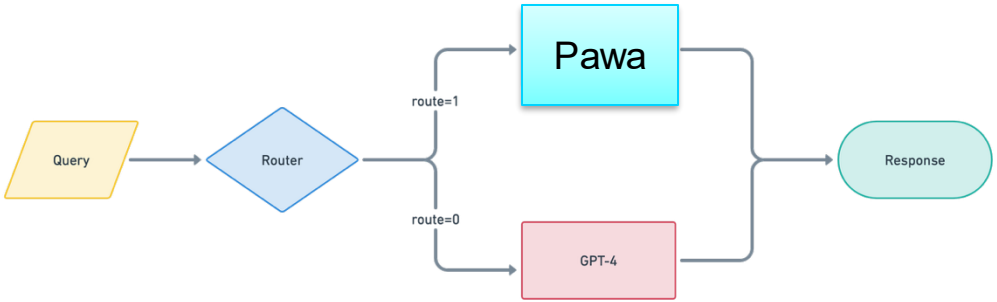
- Lower barriers to entry for AI adoption
- Democratize AI for smaller companies and developers
- Enable innovation without massive infrastructure

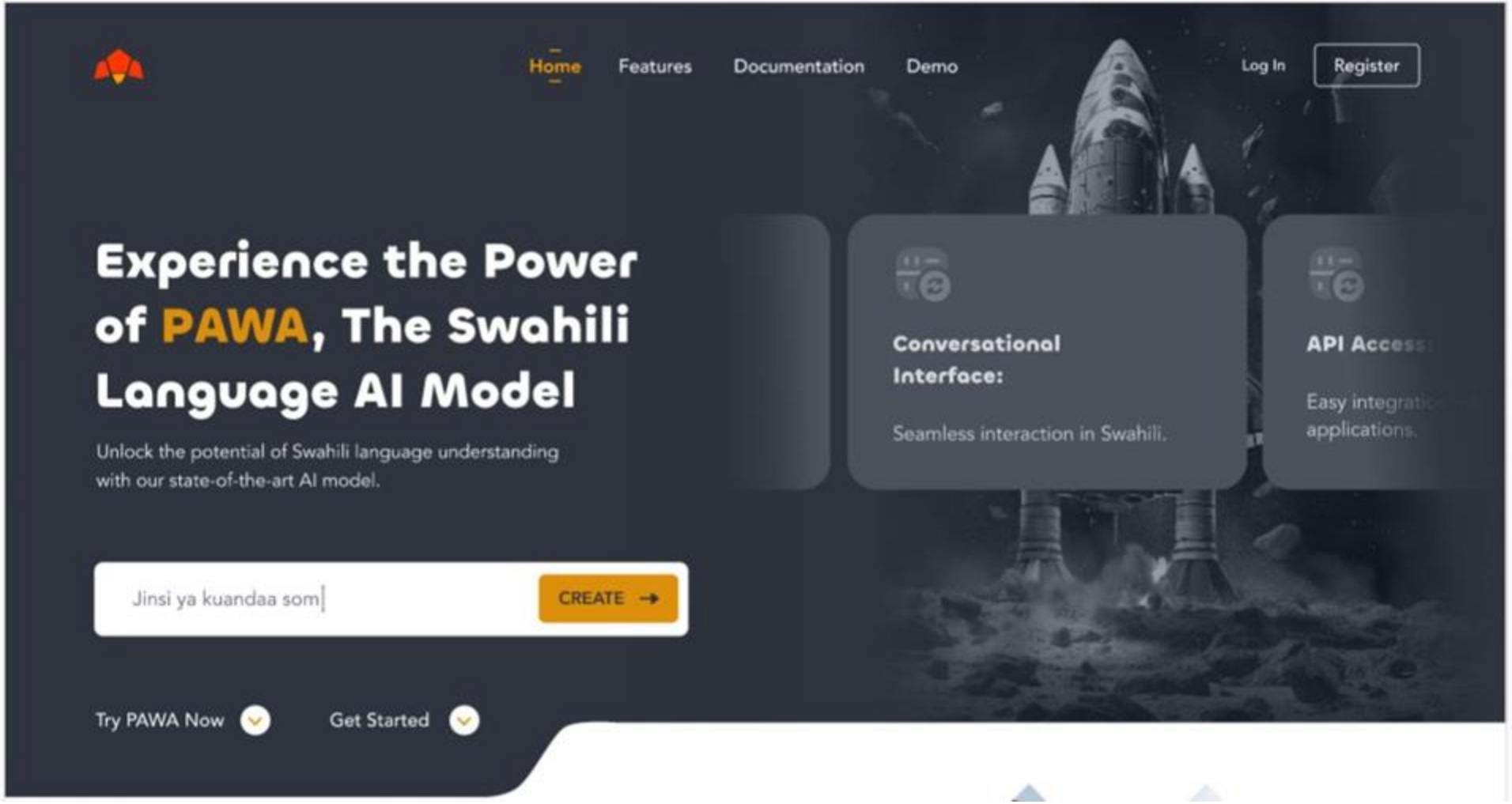
🚀 Perfect for Specific Use Cases

- Real-time applications
- Mobile and edge computing
- Privacy-sensitive environments
- Resource-constrained scenarios

🔮 The Future is Hybrid

- SLMs and LLMs will coexist
- Different models for different needs
- Specialized vs. general-purpose AI




Home
Features
Documentation
Demo
Log In
Register
Experience the Power of PAWA, The Swahili Language AI Model
Unlock the potential of Swahili language understanding with our state-of-the-art AI model.
Jinsi ya kuandaa som
CREATE
Try PAWA Now
Get Started
Conversational Interface:
Seamless interaction in Swahili.
API Access
Easy integrat
applications.

Your voice and action today define tomorrow's opportunities for millions. Join us in making this vision a reality!