



Cloud Native technologies for the AI era

Carlos Eduardo Arango Gutierrez.PhD | HPCKP-24

Upstream

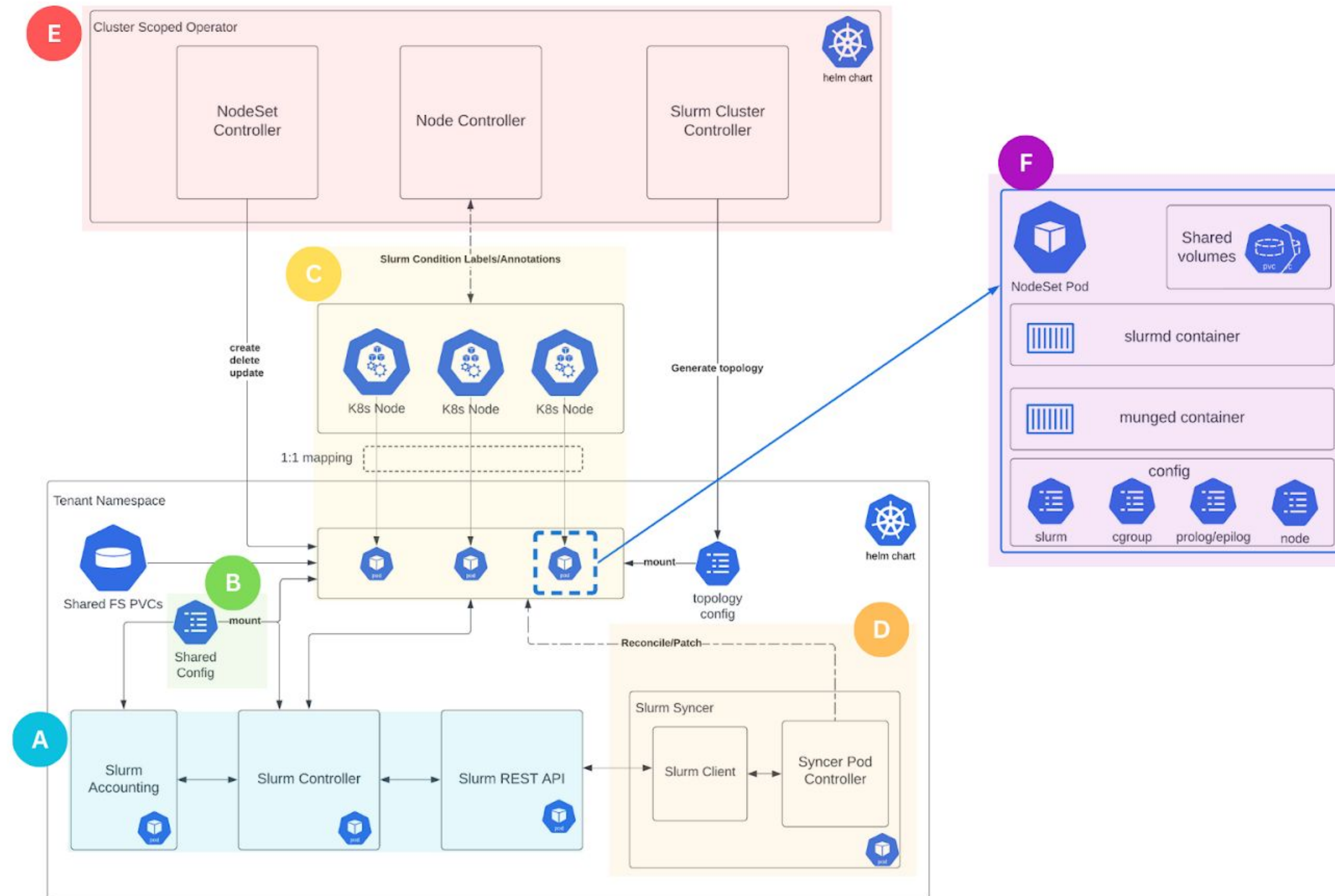


Slurm ON TOP OF K8s



Slurm ON K8s

SUNK*



source: <https://www.coreweave.com/blog/sunk-slurm-on-kubernetes-implementations>



But wait!

Who am i?

Or why do you want to hear me....



Carlos Eduardo Arango Gutierrez.PhD
Senior Systems Software Engineer @ NVIDIA

- Sylabs - Singularity Software engineer
 - RedHat - OpenShift Senior Software engineer
 - NVIDIA - CloudNative Senior System Software Engineer
-
- PhD in Computer Science focused on DevOps for HPC




Agenda

- What is the AI Life Cycle?

- Dynamic Resource Allocation (DRA)

- Kueue: Kubernetes-Native Queuing Controller

- Promises of Cloud-Native Supercomputing

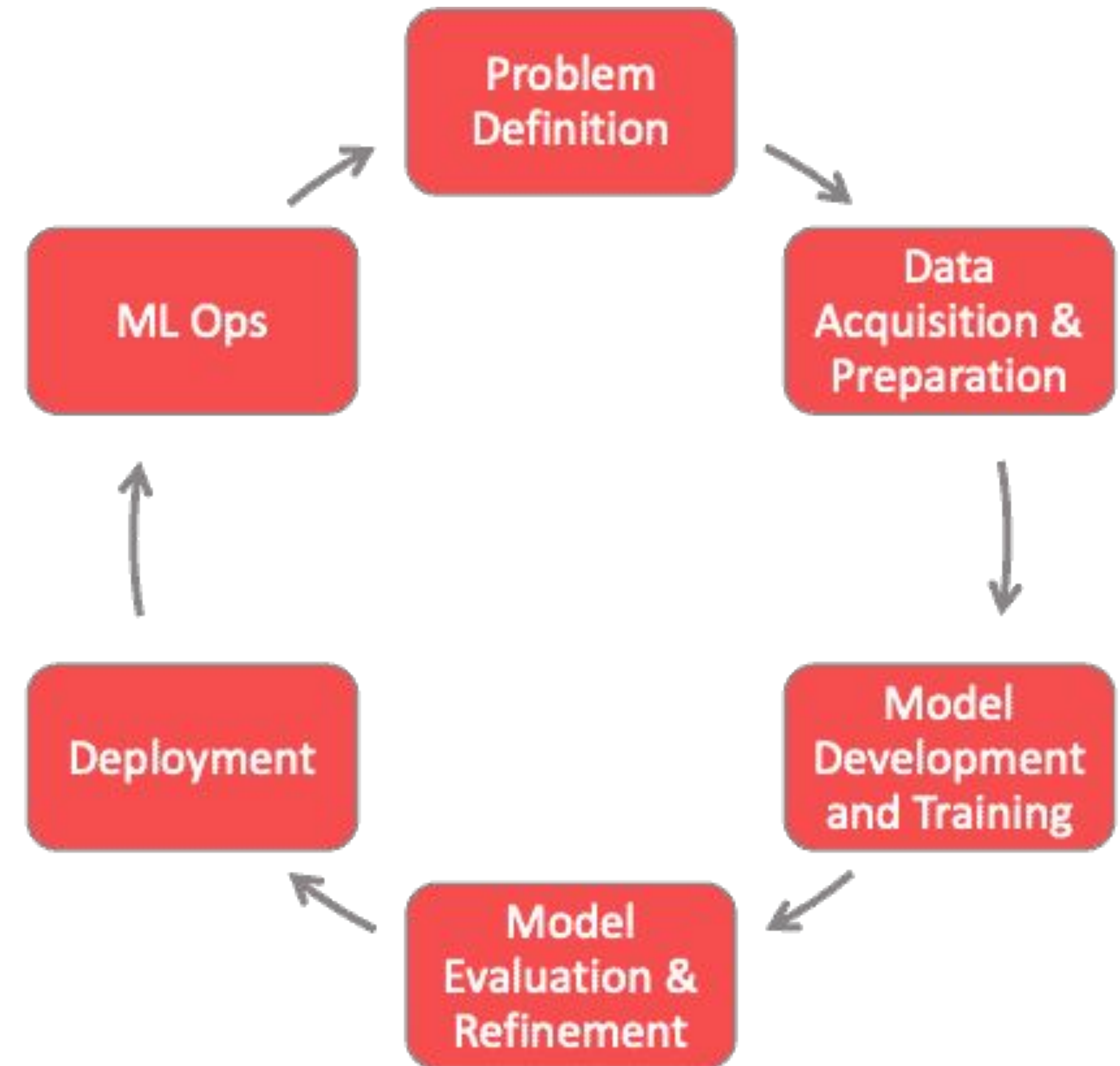


What is the AI Life Cycle?

What is the AI Life Cycle?

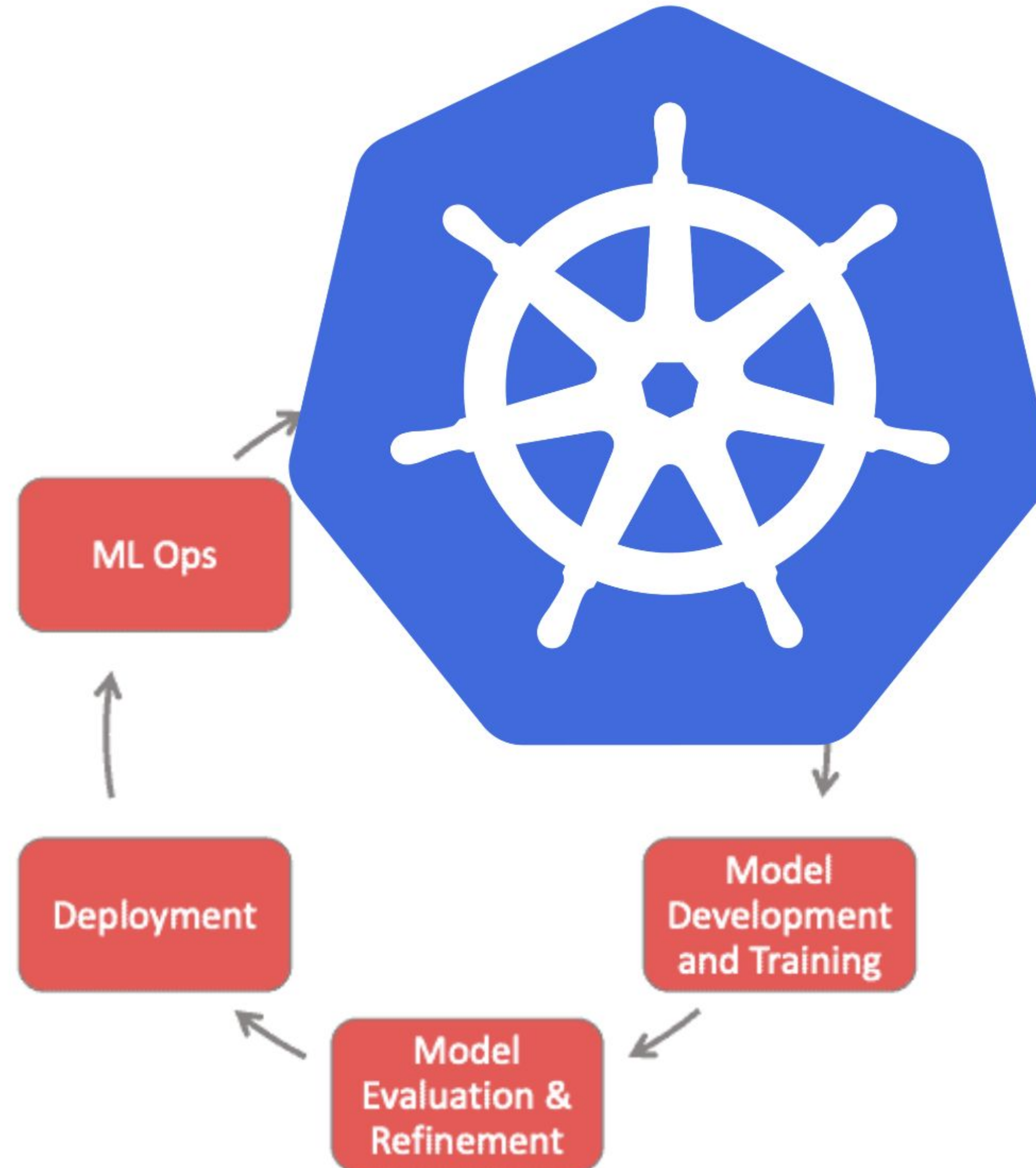
A quick 101

- Problem Definition
- Data Acquisition and Preparation
- Model Development and Training
- Model Evaluation and Refinement
- Deployment/Serving
- MLOps/Maintenance



What is the AI Life Cycle?

Where does Cloud Native play a part



The background features a complex pattern of thin, overlapping lines in shades of green and white against a black background. The lines are arranged in a way that suggests depth and movement, with some lines appearing to curve and others to intersect, creating a sense of a three-dimensional structure or a dynamic flow. The overall effect is reminiscent of a stylized, abstract representation of a network or a data stream.

Model pre-training

Model Development and Training

Model pre-training



- Kubernetes: Unmatched scalability for AI/ML pre-training.
- Automatic scaling based on demand.
- Self-healing pod lifecycle management.
- Dynamic scaling adapts to workload changes.
- Declarative approach simplifies management.
- Outperforms alternatives like Slurm for higher throughput and efficiency.



Model Training



Slurm and AI training

What Is Slurm Used For

HPC Scheduling

- The most popular scheduler for managing distributed, batch-oriented HPC workloads
- Integrates well with common HPC frameworks
- Complex to use and maintain, particularly with containerized workloads



Slurm gaps for AI training

HPC Scheduling

- Slurm's static allocation model is incompatible with the data science paradigm.
- Learning Slurm is challenging due to its complexity.
- The integration of AI with the cloud-native ecosystem is on the rise.
- Slurm Was Not Built for serving



The background features a complex pattern of thin, overlapping lines in shades of green and white against a black background. The lines are arranged in a way that suggests depth and movement, with some lines appearing to curve and others to intersect, creating a sense of a three-dimensional, crystalline or fiber-like structure. The overall effect is dynamic and futuristic.


Kubernetes!

Kubernetes (kube-scheduler)

A Cloud Native approach

- The go-to solution for flexible, containerized workloads
- Core of the cloud-native ecosystem
- Integrates well with common container-based technologies
- Requires plugins for key scheduling features of Slurm and LSF, like topology awareness and batch system capabilities



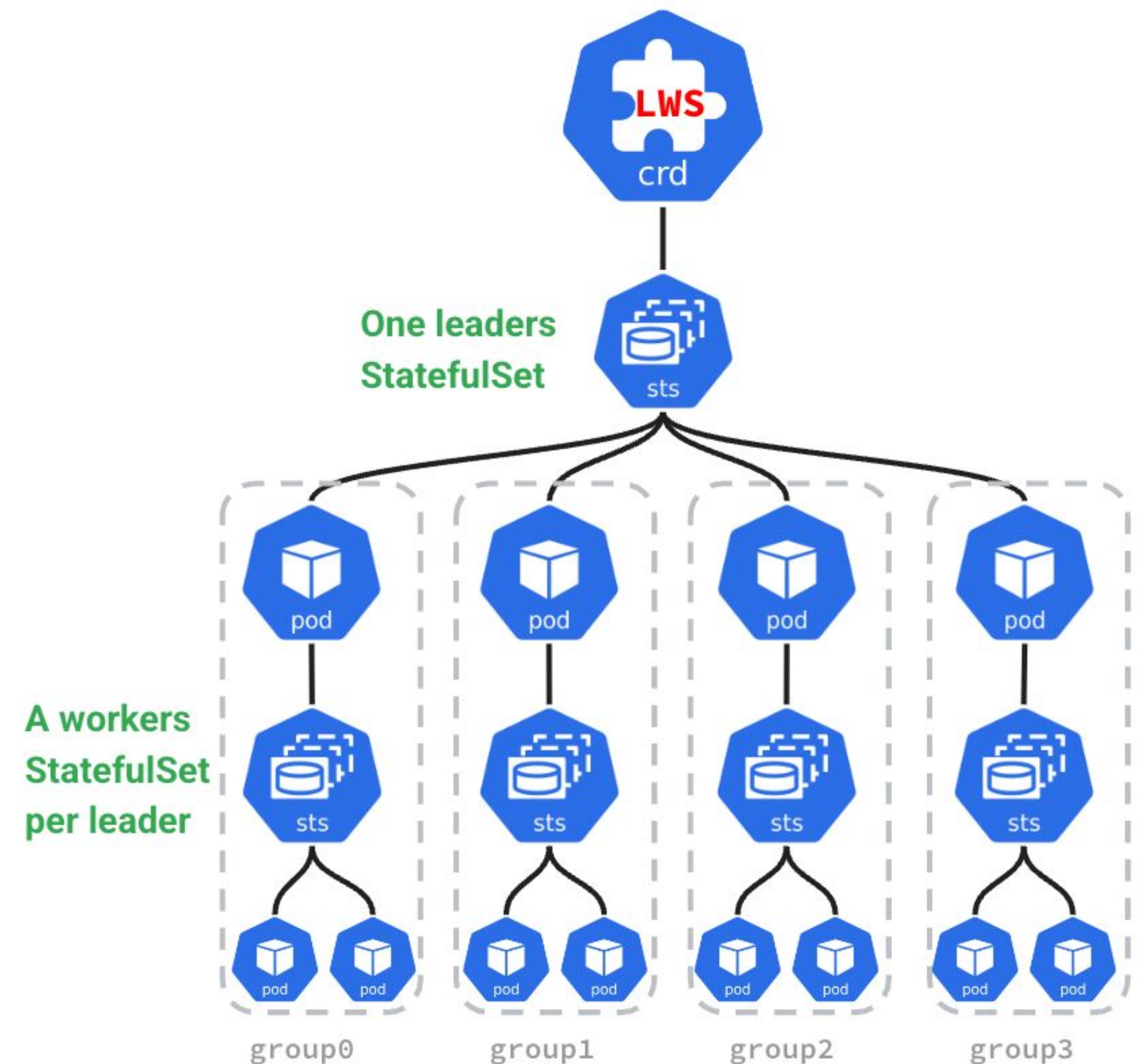
The background features a complex pattern of overlapping, semi-transparent green and white lines and streaks against a solid black background. The lines vary in thickness and orientation, creating a sense of depth and movement. Some lines are straight, while others are curved or wavy. The overall effect is reminiscent of a digital data stream or a network visualization.

Gang Scheduling: Leader Worker Set

Kueue

sigs.k8s.io/kueue

- Group of Pods as a unit
- Unique pod identity
- Parallel creation
- Dual-template, one for leader and one for the workers
- Multiple groups with identical specifications
- A scale subresource
- Rollout and Rolling update
- Topology-aware placement
- All-or-nothing restart for failure handling



The background features a complex pattern of glowing green and white lines and streaks against a black background. The lines vary in thickness and direction, creating a sense of motion and depth. Some lines are straight and parallel, while others are curved and overlapping, resembling a network or data flow visualization.

Batch scheduling:

**Kueue: Kubernetes-Native
Queuing Controller**

Kueue

sigs.k8s.io/kueue

- A job queuing operator
- Slim implementation
- Maximum reuse of core K8S
- Full compatibility with ecosystem



Kueue - Road Map

sigs.k8s.io/kueue

- Cooperative preemption support for workloads that implement checkpointing
- Flavor assignment strategies, e.g. minimizing cost vs minimizing borrowing
- Integration with cluster-autoscaler for guaranteed resource provisioning
- Integration with common custom workloads
- Budget support
- Dashboard for management and monitoring for administrators
- Multi-cluster support





GPU utilization

Slurm Generic Resources (GRES)

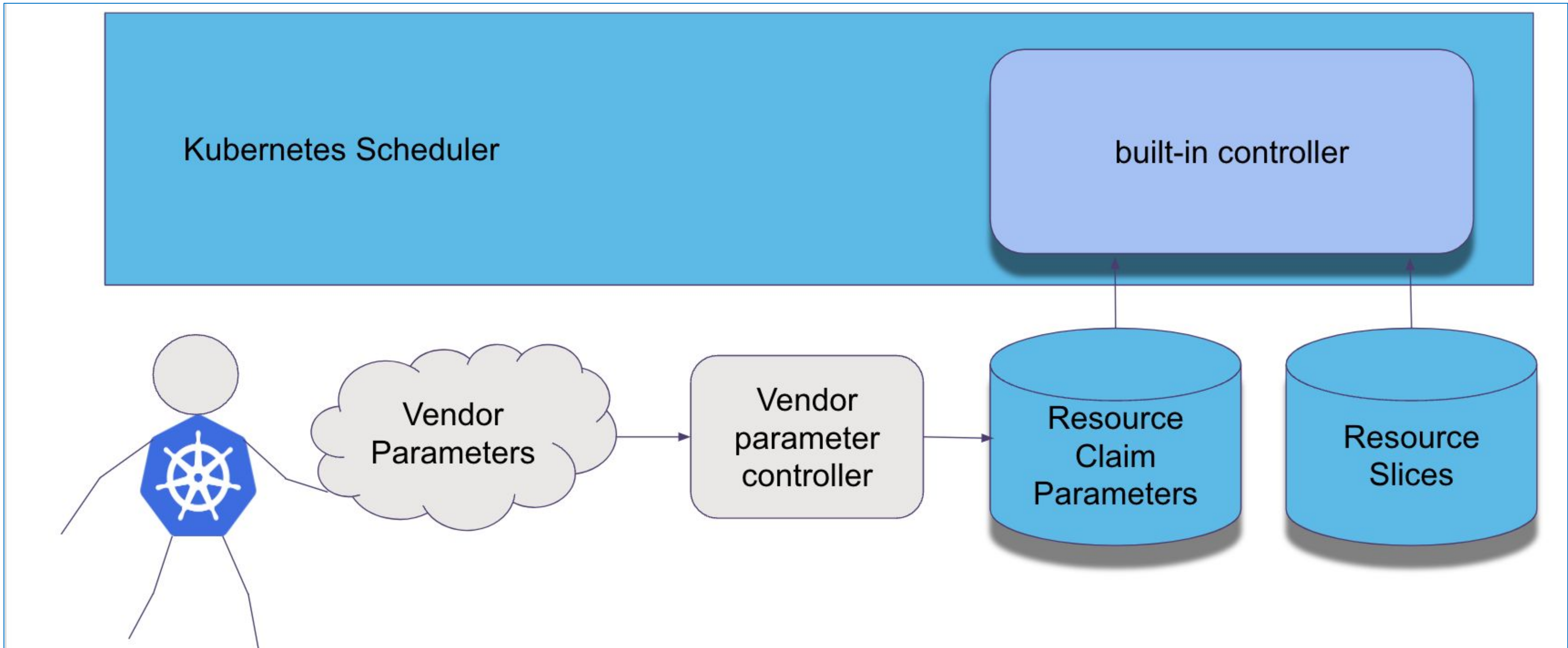
Generic Resources (GRES)

- “-gres” specifies the number of generic resources required per node
- “-gpus” specifies the number of GPUs required for an entire job
- “-gpus-per-node” same as “-gres”, but specific to GPUs
- “-gpus-per-socket” specifies how many GPUs are required per job socket (this requires that the job specifies a task socket)
- “-gpus-per-task” specifies how many GPUs are required for each task (this requires that the job specifies a number of tasks)

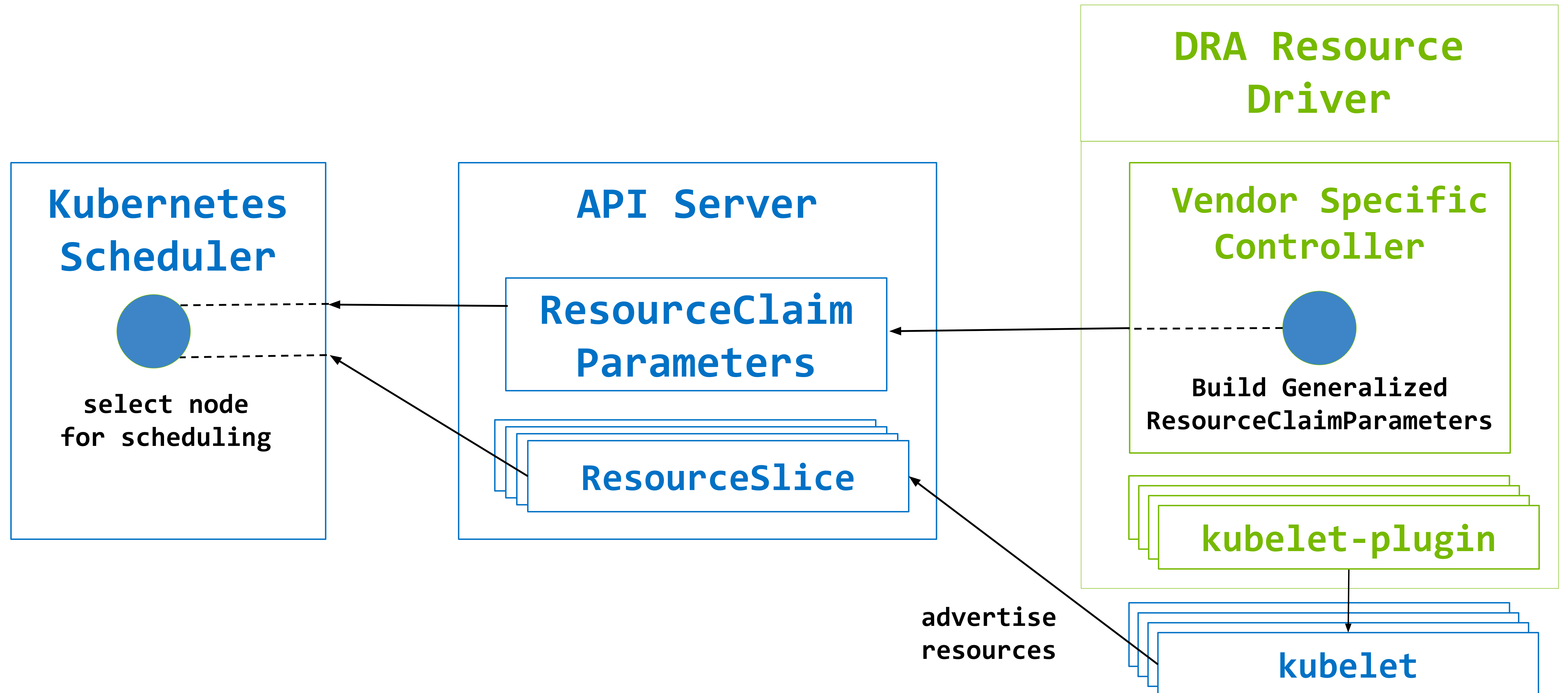
```
GresTypes=gpu,mps,bandwidth  
NodeName=tux[0-7]  
Gres=gpu:tesla:2,gpu:kepler:2,mps:400,  
bandwidth:lustre:no_consume:4G
```

Dynamic Resource Allocation

DRA!



Structured Parameters (KEP #4381): built-in parameters



The background features a complex pattern of thin, overlapping lines in shades of green and white against a black background. The lines are arranged in a way that suggests depth and movement, with some lines appearing to curve and others to be straight. The overall effect is a dynamic, almost crystalline or fiber-like structure.

Model Evaluation and Refinement

A growing landscape


It's all about going Cloud Native!

Machine Learning	Framework	Platform	Library	Framework	Platform	Library	Tool	Reinforcement Learning	Programming				
	Accord, Microsoft Lignite, ML.NET, RAY, ZenML	Angel, ForestFlow, H2O, KubeFlow, miflow	IML, mlpack, Gears, xLearn	Chainer, CNTK, TensorFlow, PyTorch, PyTorch	TensorFlow, PyTorch, jina, Polyaxon	BigDL, Catalyst, DL4J, fast.ai, Keras, TensorFlow, PyTorch	BeyondML, Intel, Intel	CleanRL, OpenAI, Google, Google	Py, Kompute, Julia, MARS, Numba, NumPy, NYOKA, PyMC3, R, SciPy, SHIP, Stan				
Data	Education	Lineage	Relational DB	Store & Format	Versioning	Operations	Feature Engineering	Stream Processing	SQL Engine	Visualization	Pipeline Management	Labeling & Annotation	Governance
	OpenDS4AI	OpenBytes, OpenLineage	CouchDB, MySQL, KV	Milvus, JanusGraph, docarray	DC, DVC, DVC	Amundsen, datashim, MARQUEZ, clairs	FEAST, feathr	Kafka, Flink, Hadoop, HDFS, HIVE, Kudu, Presto, Trino	Drill, HAWQ, Presto, SQLFlow, trino	Bokeh, Uber, D3.js, Google, Grafana, RCloud, reDash	Artigraph, Intel, DASTER, PPIW	Labelbox, Labeling, MITACHI	EGERIA
Model	Inference	Federated Learning	Training	Parameter	Format & Interface	Marketplace	Workflow	Benchmarking	Tool	Explainability	Adversarial	Bias & Fairness	
	ADUIK, XGBoost	FATE, S Substra	Horovod, Ludwig	ONNX	ONNX	Model Registry, Aquinos	Flyte, Kedro, nifi, argo, Airflow	MLPerf	FlagAI, Amazon, AWS, AWS, AWS	AI Explainability 360, ELI5	Adversarial	AI Fairness 360	
Distributed Computing	Computing & Management	Interface	Security & Privacy	Natural Language Processing	Notebook Environment								
	EDL, SOAJS, Spark, Databricks, GNS3, Netflix, Hadoop, HDFS, HIVE, Kudu, Presto, Trino	Sparklyr	Google, IBM, Microsoft, Google	DELTA, ROSS, Google, AllenNLP, fastText, flair	Elyra, colab, IPython, Jupyter, Polyglot								

The LF AI & Data landscape explores open source projects in Artificial Intelligence and Data and their respective sub-domains.

LF AI & DATA Landscape

lfaidata.foundation



TODO's of Cloud-Native for the AI era

Cloud Native is bridging the gap

AI now a need for Cloud Native

- WG-Batch ->

<https://github.com/kubernetes/community/blob/master/wg-batch/charter.md>

- WG-Serving ->

<https://github.com/kubernetes/community/blob/master/wg-serving/charter.md>

- WG-device-management ->

<https://github.com/kubernetes/community/blob/master/wg-device-management/charter.md>

