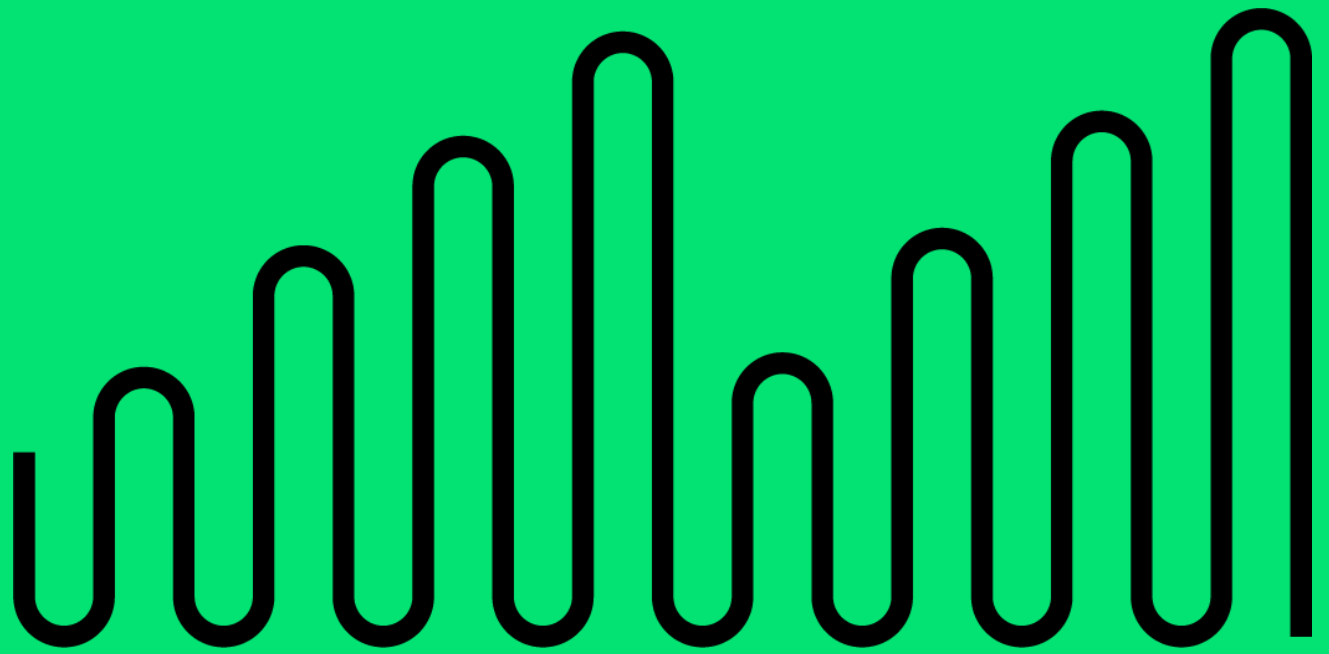
The Knowledge Hub Universities

UNIFONIC

AI Hackathon 2025

Presented by: Mahmoud Hegazy

About Unifonic



Unifonic has delivered omnichannel solutions

to over 5,000 customers in MENA.

2006
Founded

$140+M
Raised

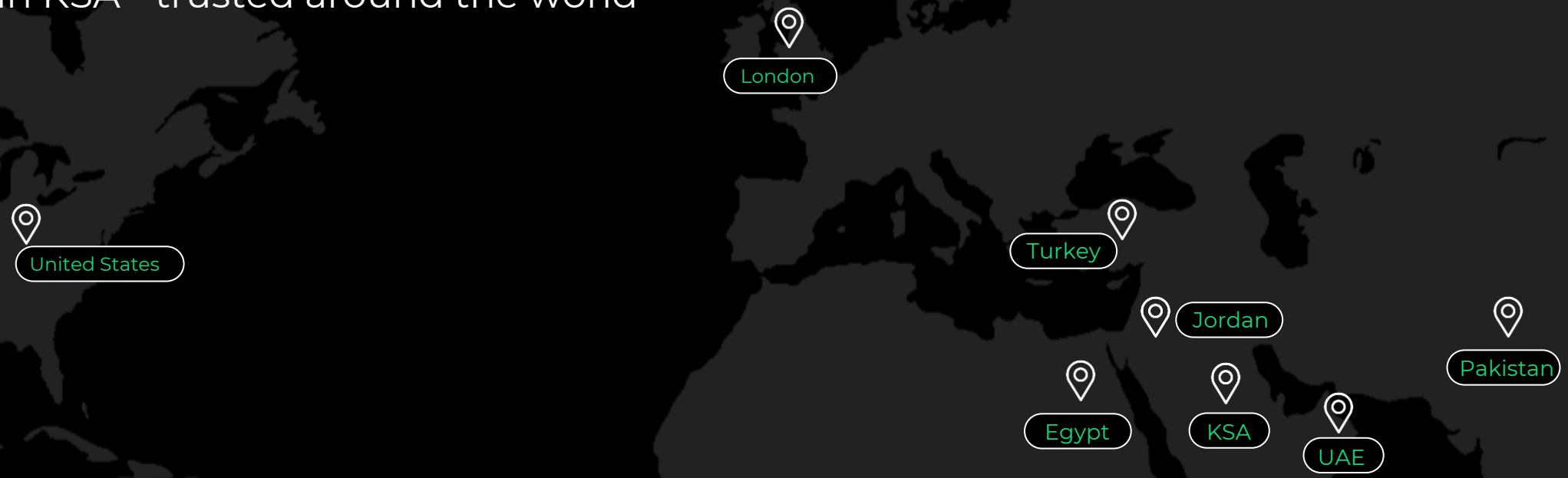
160M
End consumers across MENA

400+
Employees

10B+
Annual Transactions

Regional presence for better support.

Born in KSA - trusted around the world



Riyadh

5th floor, Hamad Tower,
King Fahd Branch Rd,
Riyadh 12212, Saudi
Arabia.

Dubai

Office # 2605, 26th Floor,
Marina Plaza Al Marsa Street,
Marina Dubai,
Dubai, United Arab Emirates.

Amman

2nd floor, Alhamdani complex,
no.141 ,Mecca Street,
Amman, Jordan.

Lahore

3rd Floor, Vogue Towers, MM Alam
Rd,
Block C2 Block C 2 Gulberg III,
Lahore, Punjab 54000 Pakistan.

Cairo

22 Desouk Street,
Off Imam Ali Street,
Ismaalyah Square, Heliopolis.

London

124 City Road,
London, EC1V 2NX,
United Kingdom.

New York

2 Park Ave 20th Floor,
New York NY 10016,
United States.

Ankara

Dumlupinar Bulvari 274/7
Mahall,
Ankara B Blok No:178 Cankaya,
Ankara.

Istanbul

Vadistanbul Bulvar Ayazaga Mahallesi
Cendere Cad. No:109B 1B Blok Ofis
No:4 / 34396 Sariyer, Istanbul, Turkey.

Islamabad

Daftarkhwan North, Plot 94
Street No. 7, Sector I-10/3,
Islamabad, Pakistan.

Let’s Start

Our Agenda

01

Introducing the Event Timeline

An overview of the 9 innovative tracks and their core AI technologies.

02

Breakdown of Each Track

A detailed exploration of the technology, research, and practical use cases.

03

Academic Papers and Innovations

Highlighting cutting-edge research and emerging trends.

04

Tools, Frameworks & Integration Strategies

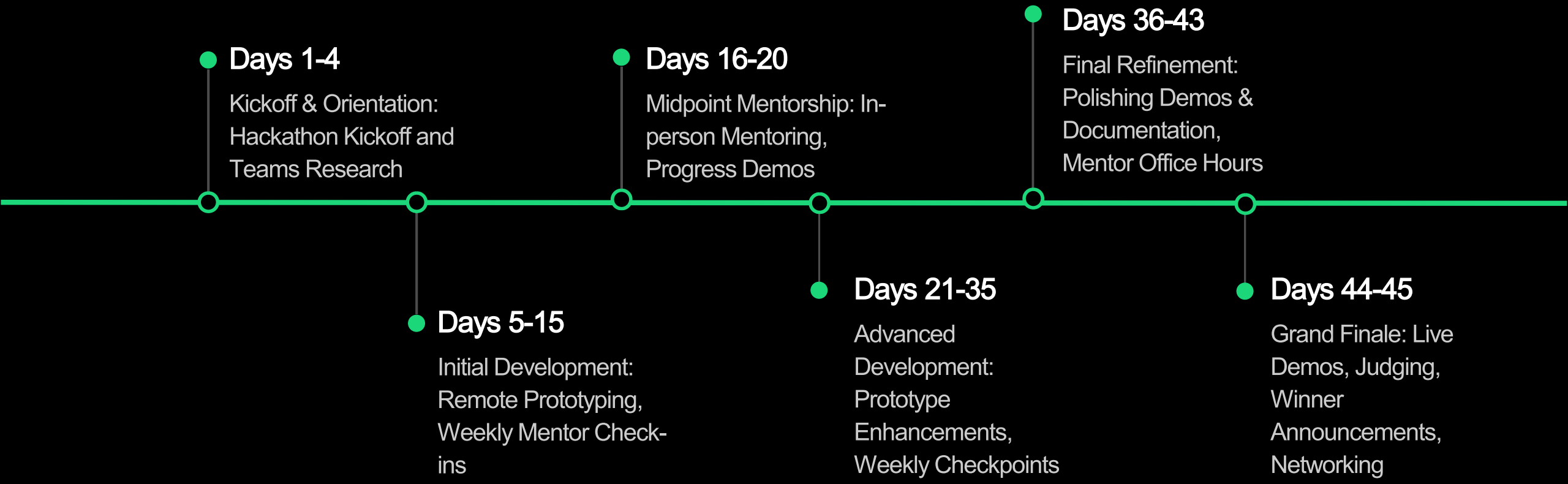
Exploring the open-source tools and frameworks that power our tracks.

05

Key Takeaways & Q&A

Summarizing insights and opportunities for future engagement.

Event Structure



Hackathon Tracks

- **AI-Enhanced DevSecOps Pipeline**
  Reduce false positives in SAST/DAST scans, auto-prioritize vulnerabilities by exploitability, and detect CI/CD pipeline anomalies with AI.

- **The Reconnaissance Crew**
  Automate the reconnaissance phase of a penetration test, including OSINT, network mapping, and vulnerability scanning.

- **The Web App Pentest Crew**
  Focus on finding vulnerabilities in a web application, including crawling, fuzzing, and exploitation.

- **Bug Triage Automation**
  Build an AI assistant that sorts and prioritizes software bugs automatically, grouping similar issues and suggesting severity levels to speed up developer response.

- **Automated Data Pipeline Validation & Monitoring**
  Create an AI system that checks data pipelines for errors or failures, monitors data quality, and alerts teams when issues appear - keeping pipelines reliable and accurate.

- **The Threat Intelligence Crew**
  Automate continuous threat monitoring, including news scraping, CVE tracking, and threat briefing generation.

- **Detecting AI Hallucinations (Arabic & English)**
  Build systems to detect when chatbots or RAG (Retrieval-Augmented Generation) systems produce hallucinated answers, such as wrong responses, out-of-domain responses, and contradictions to company knowledge/documents.

- **Harmful Text & Spam Detection**
  Develop AI models to classify text (chat, SMS, email, etc.) for spam, harassment, toxicity, and sensitive/harmful language.

- **AI-Augmented SRE for Kubernetes Applications**
  Build an AI tool that helps investigate incidents in Kubernetes apps. It should detect problems, start an automated root cause analysis (RCA), and suggest fixes to speed up recovery.

AI Powered Threat Intelligence

What is Threat Intelligence

Threat Intelligence is the collection, processing, and analysis of data about current or potential threats to an organization's digital assets. It transforms raw security data into actionable insights for detection, prevention, and response.

Key Components of Threat Intelligence

Indicators of Compromise

IPs, domains, file hashes, or patterns linked to malicious activity

Tactics, Techniques & Procedures

Behavioral insights into how attackers operate (aligned with MITRE ATT&CK)

Threat Actor Profiling

Identifying motivations, capabilities, and targets of threat groups

Campaign & Infrastructure Mapping

Discovering how infrastructure (servers, emails, malware) connects across campaigns

Why Threat Intelligence Needs AI

The Challenge

Traditional manual analysis is overwhelmed by complex modern threats like polymorphic attacks and novel evasion techniques.

The Solution

AI automates log parsing, behavioral analysis, and IOC extraction, enabling faster threat identification and response.

Open-Source Tools & Stack

MISP

For structured threat sharing and correlation; supports integration with external threat feeds.

Zeek + YARA + Suricata

For real-time packet inspection and behavioral rule matching.

TheHive + Cortex

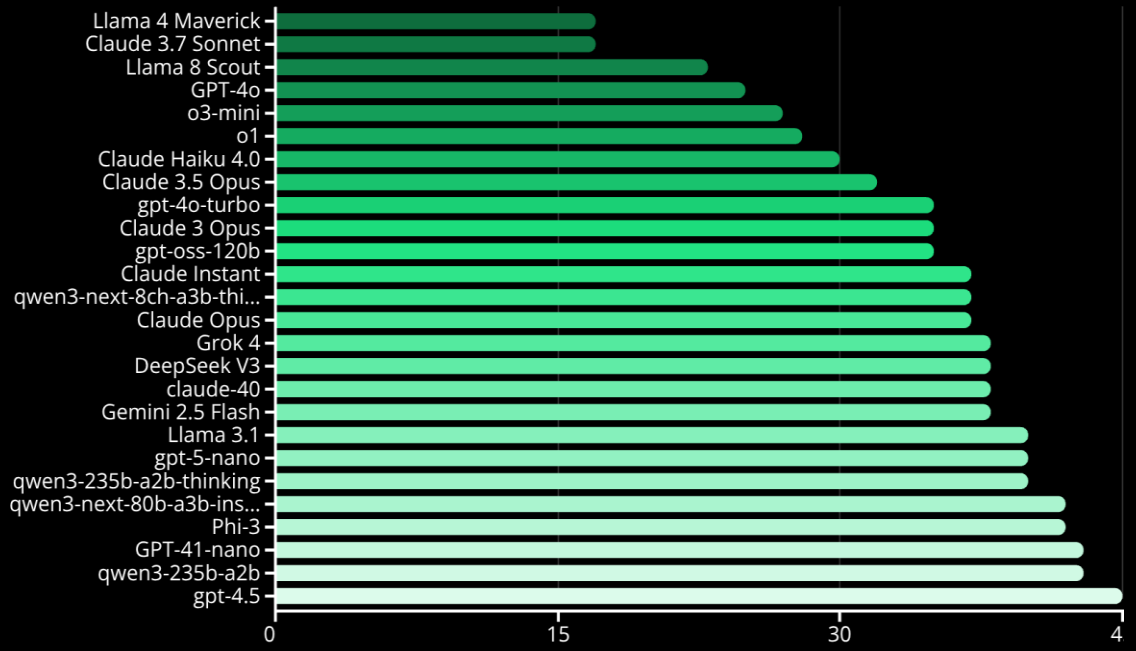
Collaboration & enrichment for incident response automation.

AI Layer

Use Python with scikit-learn or PyTorch for log anomaly detection; fine-tune BERT-like models for extracting entities from threat reports.

Detecting AI Hallucinations

Understanding Hallucinations in LLMs



This chart illustrates the varying hallucination rates of different AI models, with Llama 4 Maverick and Claude 3.7 Sonnet demonstrating the lowest rates.

What are LLM Hallucinations?

Confidently generated incorrect or fabricated information.

Risk in Security Contexts:

Poses serious risks in threat intelligence, cybersecurity, and AI safety.

Real Consequences:

Leads to misinformation, missed threats, and flawed decision-making.

Open-Source Detection Pipelines

G-Eval & Lynx

Use one LLM to evaluate another's output, especially effective in RAG setups.

XLM-R, AraT5

Pretrained multilingual models suitable for building dual-language hallucination classifiers.

Arabic Hallucination Dataset

OSACT 2024 introduced a dedicated benchmark for detecting hallucinations in Arabic language models.

HalluVerse25

A multilingual benchmark covering English, Arabic, and Turkish for evaluating hallucination detection.

Harmful Text & Spam Detection

Problem Framing

Platforms need to filter hate speech, phishing, fraud attempts, and AI-generated spam across languages. Challenges include detecting multilingual content, code-mixed slang, and evolving adversarial phrasing that evades traditional keyword-based filters.

Tools & Models

Detoxify & Perspective API

Transformer-based toxicity scoring in real-time

HateBERT, Spam-T5

Fine-tuned models for Reddit-based toxicity and email spam classification

Apache SpamAssassin + ML

Hybrid model for spam detection using AI-enhanced feature extraction

Multilingual Models

Models like XLM-R and AraT5 enable detection of harmful text across languages

AI Powered K8s SRE

What Is Site Reliability Engineering (SRE)?

Site Reliability Engineering (SRE) is a discipline that applies software engineering principles to improve the reliability, scalability, and performance of infrastructure and services. When applied to Kubernetes environments, SRE becomes essential due to the platform's inherent complexity.

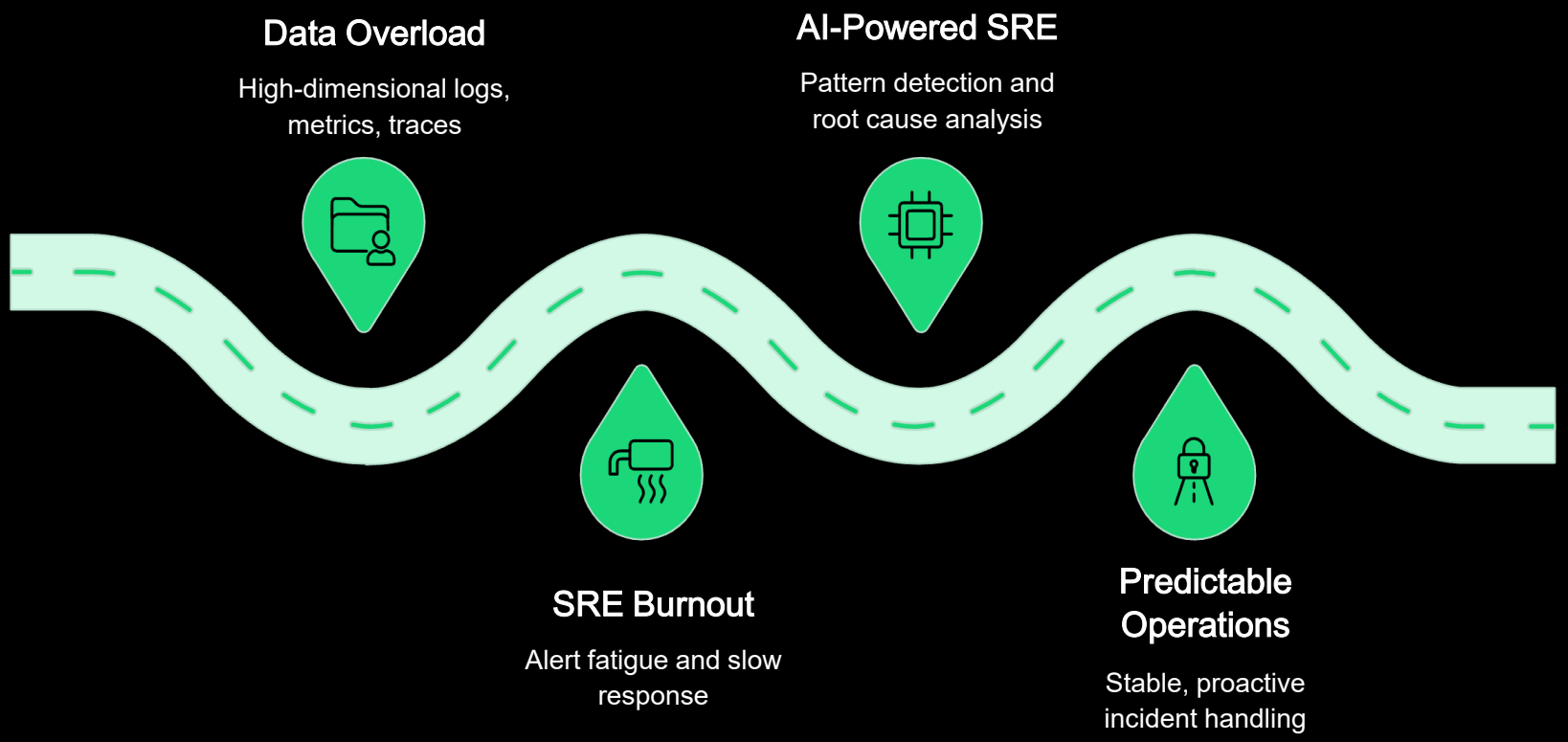
What Is Kubernetes SRE?

Focus

Ensures Kubernetes-based applications run reliably in production, with minimal downtime and operational overhead.

Core Responsibilities

- Monitor clusters and services for health and performance
- Automate deployment rollouts and rollbacks
- Detect and resolve incidents (e.g., crash loops, failed pods, network bottlenecks)
- Maintain SLOs/SLAs (latency, uptime, error budgets)
- Coordinate post-incident analysis and continuous improvement



1

The Challenge: Data Overload

High-dimensional telemetry data (logs, events, metrics, traces) overwhelms traditional SRE practices due to its sheer complexity.

2

The Problem: SRE Burnout

This leads to alert fatigue, slow incident response, and inefficient operations, impacting system reliability and team well-being.

3

The Solution: AI-Powered SRE

AI-powered observability enables pattern detection, anomaly scoring, and root cause analysis for more predictable and self-tuning Kubernetes operations.

Traditional SRE vs AI Powered SRE

Characteristic	Data Analysis	Alerting	Incident Response	Insights	Capacity Planning
Traditional SRE	Manual analysis of logs, metrics, and alerts	Reactive alerting based on predefined thresholds	Slow incident response due to manual correlation	Lack of proactive insights into potential problems	Difficult capacity planning without advanced analytics
AI-Powered Observability	Automated analysis using machine learning	Intelligent alerting with noise filtering and prioritization	Faster incident response with automated root cause analysis	Predictive analytics for proactive problem prevention	Improved capacity planning with machine learning models

Toolchain & Architecture

Prometheus + Grafana

Prometheus is the de-facto metrics collector in cloud-native stacks and can be paired with AI/ML services for anomaly detection.

ELK/OpenSearch Stack

The ELK/OpenSearch stack offers machine learning plugins to detect outliers in log data and do smart alerting.

Event Correlation Engines

Event correlation engines like MozMixer can reduce alert noise by grouping related Kubernetes events, with an AI layer to identify non-obvious linkages.

AIOps Platforms

Open-source AIOps platforms like StackStorm or Apache Airflow can orchestrate automated incident responses triggered by anomaly detectors.

K8sGPT

K8sGPT is an AI co-pilot for cluster diagnostics that integrates Kubernetes metadata with large language models.

AI-Enhanced DevSecOps Pipeline

What is CI/CD

CI/CD (Continuous Integration / Continuous Delivery or Deployment) is a software development practice that automates the process of building, testing, and releasing applications.

- CI – Continuous Integration:
  Developers frequently merge code changes into a shared repository. Automated tests run to ensure the code works together smoothly.
- CD – Continuous Delivery/Deployment:
  Automatically prepares (Delivery) or releases (Deployment) applications to production with minimal manual steps.

Goal: Faster releases, higher quality, fewer errors.

Enhancing Security in CI/CD Pipelines with AI

Security often bottlenecks in CI/CD pipelines

Manual scanning creates alert fatigue, slowing down the development process

AI helps prioritize exploitable risks

AI models can analyze code and vulnerabilities to identify the most critical issues that require immediate attention

AI flags insecure code patterns

Machine learning algorithms can detect known vulnerabilities and insecure coding practices, enabling developers to address them early in the development cycle

AI proposes remediations

AI-powered tools can suggest specific fixes and mitigations for identified security vulnerabilities, accelerating the remediation process

OSS Stack for CI/CD Security

Semgrep

Lightweight static analyzer with AI-generated rules for detecting security vulnerabilities in code.

Trivy

Open-source scanner for finding vulnerabilities in containers, infrastructure as code (IaC), and software bill of materials (SBOMs).

DefectDojo

Vulnerability aggregation and deduplication platform that can be integrated with AI-powered security tools.

LLM-Powered Merge Request Comments

Use large language models to automatically generate suggestions and explanations for security issues directly in merge requests.

The Reconnaissance Crew

The AI-Driven Recon Landscape

The Rise of Automated Recon

Showcase how threat actors are increasingly using automated tools and techniques to rapidly identify and exploit vulnerabilities, forcing defenders to match their pace.

AI-Powered Reconnaissance

Illustrate how AI enables the automation of asset discovery, metadata scraping, and weak point identification across a vast attack surface.

Addressing the Recon Challenge

Demonstrate how AI-driven reconnaissance tools can help defenders keep up with the rapid pace of threat actors and automate key reconnaissance tasks.

Tools for Recon Automation

SpiderFoot

Automated data gathering tool for OSINT and infrastructure reconnaissance. Collects information from a wide range of sources to build a detailed profile of target assets.

Recon-ng

A feature-rich CLI framework for conducting reconnaissance. Supports scripting and integration with AI-powered modules for efficient data gathering and analysis.

Amass

Advanced passive and active subdomain discovery tool. Leverages multiple data sources and techniques to identify potential attack surfaces in the target's infrastructure.

Pinecone + GPT-4

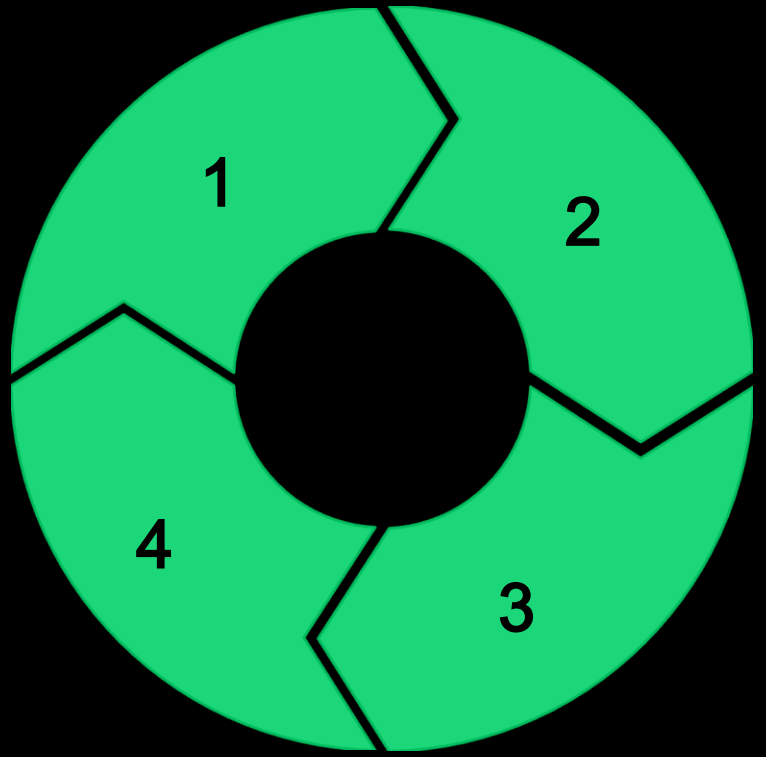
Stores reconnaissance data as embeddings in a vector database and allows natural language querying to uncover insights, such as identifying hosts exposing admin panels.

The Pentesting Crew

AI Assisted Pentesting

Reconnaissance

AI automates information gathering, identifying targets and data points.



Vulnerability Identification

AI scans for and flags potential weaknesses and security gaps.

Reporting

AI compiles findings, prioritizing risks and suggesting remedies.

Exploitation

AI agents simulate attacks to test and confirm vulnerabilities.

This AI-driven cycle streamlines the pentesting process, allowing human experts to focus on complex analysis and novel threat discovery.

OSS Tools for Hacking with AI

PentestGPT

Framework for automated exploit generation and execution, leveraging GPT-4 for tasks like recon, injection, and exploit chaining.

AutoPentest Toolkit

Aggregates various AI-augmented scanning and fuzzing tools, automating different stages of the penetration testing process.

AI-Generated Payloads

LLMs can mutate and generate XSS, SQLi, and other malicious inputs based on the observed behavior of web application firewalls (WAFs).

AI-Written Pentest Reports

LLMs can save time by automatically summarizing and documenting the findings from a penetration test.

Bug Triage Automation

Automating Bug Triage

Turning Chaos Into Clarity

The Challenge

- Hundreds of issues weekly
- Manual prioritization by severity, impact, duplication
- Time-consuming assignment process
- Growing backlog & bottlenecks

The Solution

- AI-powered instant prioritization
- Automated, smart assignment
- Significantly reduced backlog
- Enhanced team efficiency & focus

Open Source Tools & Model Choices

DeepTriage

Transformer model that predicts labels, owners, and duplicate likelihood for bug reports.

CodeBERT

Fine-tuned on structured bug data for classification tasks like component prediction and severity labeling.

RoBERTa

Another fine-tuned model suitable for bug triage, leveraging the power of self-supervised pretraining on large corpora.

Pipeline: GitHub Webhook → API Gateway → LLM Classification

Automated pipeline to ingest new issues, classify them using LLMs, and auto-assign labels and owners.

Automated Data Pipeline Validation & Monitoring

What Is Data Validation and Why It Matters

Definition & Impact

- Silent data corruption
- Broken dashboards
- Model drift and failure
- Missed revenue/insights

Common Validation Checks

- Null or missing value thresholds
- Schema adherence (column data types)
- Value ranges and distribution integrity
- Referential integrity (foreign keys, joins)

Data Pipelines Without AI: The Risk

1

Manual Rule-Based Monitoring

Hardcoded assertions (e.g., "column X must be non-null")

Static thresholds that don't adapt to seasonality or variance

Alert noise: Too many false positives or missed anomalies

2

Operational Risks

Pipelines silently break due to upstream system changes (schema drift)

Unexpected volume shifts

Inconsistent source behavior

3

Real Failure Scenarios

Marketing dashboard shows zero revenue due to null product IDs

Model trains on mislabeled dataset due to upstream ingestion errors

Reported KPIs fluctuate erratically due to timezone offset bug

Tools for Observability & ML Monitoring

Great Expectations (GX)

An open-source framework for defining and validating data quality expectations, supporting schema, null, and range checks.

Deequ (AWS Labs)

A library built on Apache Spark for computing data quality metrics, detecting data distribution anomalies, and performing statistical drift analysis.

PyOD, TensorFlow Probability

Python libraries for building custom anomaly detection models using unsupervised machine learning techniques like isolation forests and probabilistic models.

Pinecone

A vector database that can store data pipeline metrics and telemetry as embeddings, enabling fast retrieval and querying using LLMs.

What Winning Projects Include

Realistic problem framing with measurable output

Projects should clearly define the problem they are addressing and outline how their solution can be evaluated and measured for impact.

Integration of multiple tools in a coherent architecture

Winning projects will demonstrate the ability to leverage various open-source tools and frameworks, integrating them into a cohesive and scalable solution.

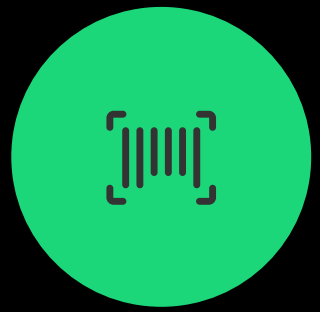
Evidence of evaluation (metrics, before/after)

Projects should provide quantitative and qualitative data to support the effectiveness of their solutions, such as performance metrics and comparative analyses.

Judging Criteria

Innovation (30%)

Creativity and novelty of the proposed solution



Technical Execution (30%)

Functionality, feasibility, and robustness of the prototype

Impact (20%)

Relevance and potential to address real-world AI security/responsibility challenges

Presentation (20%)

Clarity of explanation, quality of live demo, and overall storytelling

Questions?