Kara Thrace - CV

Kara Thrace
Senior Data Engineer | “If it can be broken, I’ll fix it—if it works, I’ll break it first.”
Location: Somewhere between Caprica and a Kubernetes Cluster
Email: kara.starbuck@datapipelines.com | Phone: +1010-0101-HARPY

PROFILE

Rebellious, resilient, and resourceful Data Engineer with a reputation for turning chaos into functioning systems.
Starbuck thrives in environments where nothing goes as planned—when clusters are overloaded, budgets cut,
and infrastructure is falling apart, that’s where she shines. She has extensive experience building battle-tested
data pipelines that continue running under extreme conditions such as outages, sabotage, or outright explosions.
Known for rejecting unnecessary bureaucracy and replacing it with practical, hands-on solutions, she is
unconventional but consistently delivers reliable results. For organizations facing impossible deadlines and
unforgiving environments, Starbuck is the engineer who will make sure the job gets done, even if it means
rewiring the system mid-flight.

TECHNICAL SKILLS

- Data Pipelines & Orchestration: Apache Airflow, dbt, Apache NiFi, Kafka, duct-tape-based monitoring systems
- Languages: Python, SQL, Go, Bash, fluent in “military sarcasm”
- Cloud & Platforms: Snowflake, AWS, Azure, Cylon-resistant on-prem clusters
- Specialties: Chaos engineering, schema recovery, anomaly detection under fire, distributed system triage
- Soft Skills: Battle leadership, sarcastic communication, poker-based estimation, decision-making under duress

EXPERIENCE

Battlestar Galactica Fleet – Lead Data Engineer (4 years)
• Designed and maintained real-time telemetry pipelines for Viper flight data, weapons systems, and ship sensors.
Enabled commanders to detect Cylon incursions in seconds rather than minutes, saving countless lives.
• Developed anomaly detection models using streaming data, reducing false negatives in raider detection by 87%.
• Optimized Spark jobs to run in environments with severe resource constraints, often doubling performance
despite limited compute power and failing nodes.
• Created a resilient CI/CD system from scratch without external package registries or internet access, ensuring
deployments continued across a fractured fleet network.
• Championed resilience-first engineering by deliberately introducing failures into systems to test recovery paths.
Documented chaos scenarios became training exercises for junior engineers.

Caprica Ops – Junior Data Wrangler (Before the Fall)

- Ingested telemetry from Viper squadrons into centralized databases, normalizing messy logs into structured datasets that informed pilot training and performance reviews.
- Built predictive models for pilot readiness and mission risk scoring, later adapted into informal poker odds calculators used off-duty (with surprisingly accurate results).
- Responsible for maintaining relational databases supporting mission analytics, learned the hard way about the consequences of schema drift and unindexed joins.
- Accidentally crashed the primary Caprica cluster during a “tuning exercise,” but successfully covered it up as a suspected cyber-attack—gaining her reputation as someone who “fights Cylons in SQL.”

PROJECTS

Cylon Detector v2.0

- Problem: Command needed an early-warning system for identifying raider incursions based on ship telemetry.
- Solution: Built a Kafka-based pipeline to classify normal vs hostile activity, applying anomaly detection in real time.
- Result: Achieved 99% detection accuracy, although several false positives were triggered by coffee machine electrical interference. The project saved lives, boosted morale, and proved the value of predictive data systems.

FTL Jump Logger

- Problem: Fleet jumps lacked consistent record-keeping, leading to drift errors and lost ships during hyperspace travel.
- Solution: Designed a distributed, redundant system to log every FTL jump, capturing latency, drift data, and ship metadata for after-action analysis.
- Result: Standardized reporting across the fleet, creating the first-ever reliable jump audit trail in colonial history.

Fleet-Wide Data Mesh

- Problem: Commanders demanded autonomy over data systems, resulting in silos and duplication across ships.
- Solution: Initiated a data mesh governance model to federate responsibility, supported by lineage-tracking tools.
- Result: Collapsed in political infighting and sabotage after three weeks, but produced documentation that later informed fleet-wide data standardization efforts. Privately considered a personal win because she bet against it.

MISSION LOGS (Sample Jira Tickets)

- Ticket #421: Pipeline outage traced to a nuclear detonation adjacent to main cluster.
  Resolution: “Avoid future nukes.”
- Ticket #422: Schema drift detected after Viper firmware 2.7 upgrade introduced undocumented fields.
  Resolution: Patched dbt models manually mid-battle; flagged manufacturer for poor release notes.
- Ticket #423: ETL jobs failing due to anomalous query load traced to Cylons attempting SQL injection.

Resolution: Blocked malicious queries, deployed firewall rules, and publicly mocked the attackers in commit notes.
- Ticket #424: Fleet-wide data warehouse unavailable due to unexpected power draw from FTL systems.
Resolution: Rebalanced workloads, then rerouted pipelines through backup servers in auxiliary hangar.
- Ticket #425: Analyst complaint: “Why are there jokes in the logs?”
Resolution: “Because if you don’t laugh, you’ll cry.”

EDUCATION
- Colonial Flight School (Graduated “mostly alive”): Gained deep appreciation for resilience, adaptability,
and the need for fault-tolerance in every system—whether in a cockpit or in code.
- Self-taught in SQL and Python during late nights between missions, later trained junior engineers and pilots
on database basics and data literacy.

INTERESTS
- Enjoys cigars, poker, heavy drinking, and inventing new chaos scenarios to “stress test” both humans and systems.
- Known for deliberately breaking stable systems just to observe the recovery process and improve documentation.
- Advocates for “fun in engineering,” insisting morale is a core component of system reliability.