

Domain(keep close to resumes)	Technical Challenge
AI/ML & Smart Manufacturing	<p><b>Problem:</b> You are deploying a defect detection model (ResNet-50) on legacy Industrial PCs with only 512MB RAM total. The OS consumes 400MB. The Python inference script crashes with an Out-Of-Memory (OOM) error because loading the model and intermediate activation maps requires ~300MB.</p> <p><b>Task:</b> Refactor the inference pipeline to fit within a &lt;100MB heap budget. Explain how you would use Image Tiling (processing 256x256 chunks), INT8 Quantization, and aggressive memory freeing of activation tensors to prevent the crash.</p>
	<p><b>Problem:</b> You are coding firmware for a disposable \$2.00 Smart Patch with a microcontroller that has only 4KB of SRAM. You must store a 24-hour temperature history (sampled every minute = 1,440 readings) to transmit via Bluetooth. Storing these as standard float (4 bytes) would take ~5.7KB, causing a stack overflow.</p> <p><b>Task:</b> Design a memory-efficient data structure to store this history in under 1KB of RAM while maintaining 0.1°C precision. Propose a solution using Delta Encoding or Run-Length Encoding (RLE) to pack the data.</p>
Embedded Systems & Medical IoT	<p><b>Problem:</b> A stream processing system tracks 500,000 trucks against 100,000 complex "High Risk" geofence polygons. Loading all 100,000 polygons into the RAM of the worker nodes causes the application to crash (OOM). You cannot query the database for every GPS point due to latency requirements.</p> <p><b>Task:</b> Architect a Probabilistic Filtering mechanism. Describe how to implement a Spatial Bloom Filter or a GeoHash Grid in memory to filter out 99% of safe traffic using only bits, fetching full polygon geometry from the DB only when a "probable" hit occurs.</p>
Large Scale Data & Logistics	<p><b>Problem:</b> A fire alarm control panel uses HTTP POST over Wi-Fi to send heartbeats. You need to add 4G Cellular redundancy. The current microcontroller crashes because initializing two separate TLS (SSL) stacks (one for Wi-Fi, one for Cellular) consumes 40KB of heap, exceeding the RAM limit. HTTP is also causing "Heap Fragmentation" due to constant connection setup/teardown strings.</p> <p><b>Task:</b> Architect a "Lazy-Loading" Failover Mechanism using MQTT.</p>
IoT Networking & Critical Infrastructure	<p>1. The Catch (HTTP vs MQTT): Demonstrate how switching to MQTT with a persistent connection reduces overhead by 90% (headers vs payload).</p> <p>2. The Optimization: Implement a single Shared Ring Buffer for outgoing messages. Write a state machine that only initializes the Cellular/TLS context after 3 failed Wi-Fi retries (destructing the Wi-Fi context first to free RAM).</p> <p>3. QoS Catch: Explain why you would choose MQTT QoS 1 (At least once) over QoS 2 (Exactly once) to save packet storage RAM.</p>
	<p><b>Problem:</b> You are testing a "Image Resizer" microservice used by 10M users. It accepts a URL, fetches the image, resizes it, and returns it. You suspect SSRF (Server-Side Request Forgery).</p> <p>1. The Find (Bypass): The developer blacklisted 169.254.169.254 (AWS Metadata IP) and localhost.</p> <p><b>Task:</b> Demonstrate how to bypass this using DNS Rebinding or Alternative IP Encodings (e.g., Octal: 0251.0376... or Ox7f000001) to trick the validator but still hit the internal Metadata service to steal AWS keys.</p> <p>2. The Fix (At Scale): You cannot manually patch the code of 500 different microservices that might use this library.</p>
Cloud Security & AppSec	<p><b>Task:</b> Design a "Secure-by-Default" Network Architecture. Explain how you would use a Service Mesh (e.g., Istio/Linkerd) or Egress Filtering to block all outbound connections from the container to private IP ranges (RFC1918), regardless of what the code tries to do.</p>

<b>Technical Challenge</b>	
Technical Report (2–3 pages)	Mandatory
Codebase (ZIP file or GitHub link)	Mandatory
Output/Result Files (graphs, charts, simulation results, screenshots, etc.)	
Video Demo (2–4 minutes)	
Optional: Presentation Slides (max 5 slides)	
Optional: Design Sketches or UI Mockups	