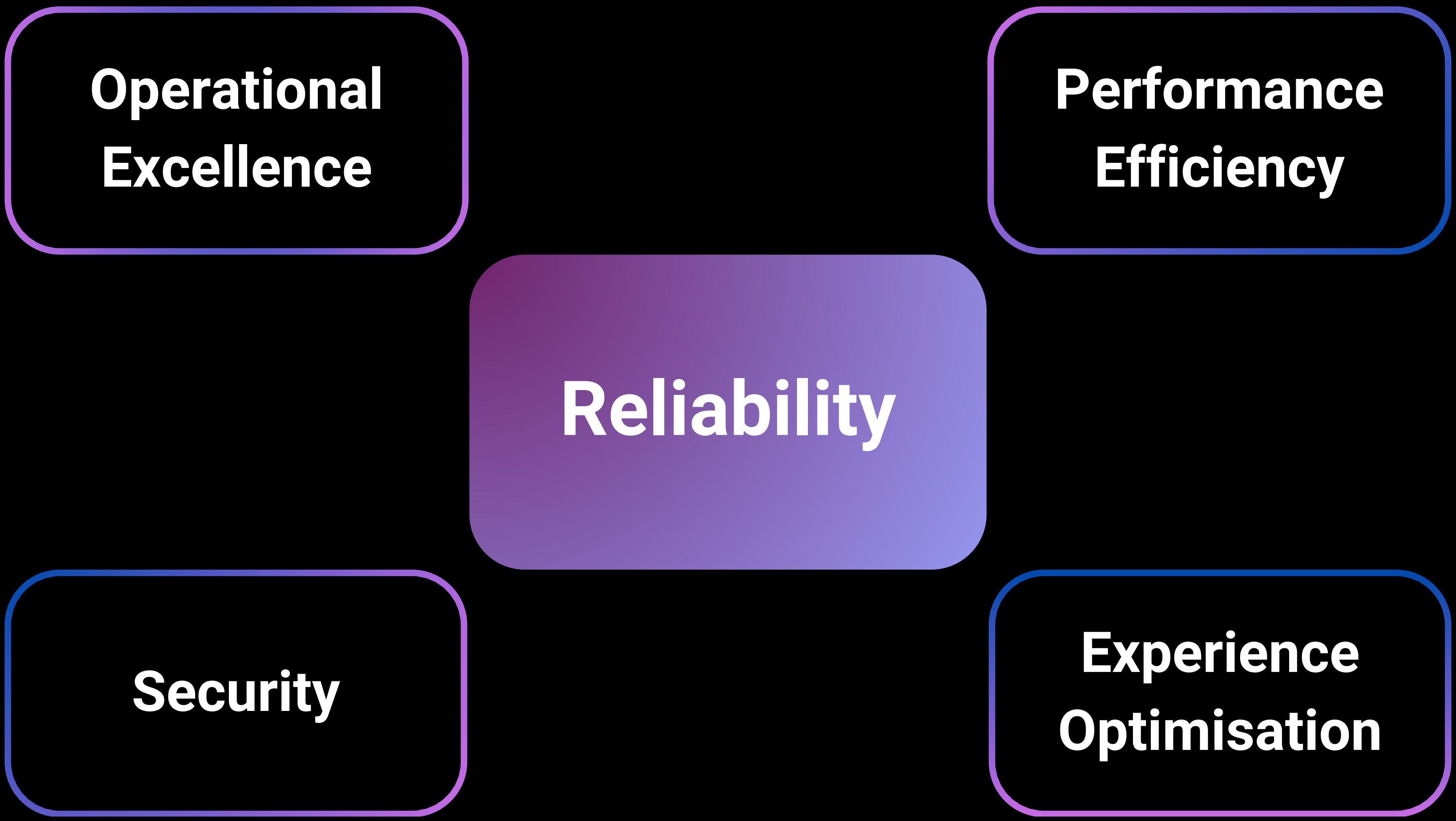
Power Platform Well Architected

Design Principles



Operational Excellence

Performance Efficiency

Reliability

Security

Experience Optimisation

nati-turtledove.com
@NatiTurts

Build solutions that can handle errors and failures. Include try-catch operations in workflows and transactions to provide fallbacks and preemptive failure responses with reduced functionality instead of user-facing errors. Ensure you also have SOPs and documentation for failure recovery, and consider monitoring with audit or log tables.

Design For Business Requirements

Try your best to gather and document in detail business requirements. This includes user experience, data needs, process workflows, and personas.

- Set measurable success metrics and goals (e.g., system flow reliability).
- Review potential limits such as quotas, capacity restraints, and regional features.
- Identify dependencies to prevent chain reaction failures and deployment errors.

Keep requirements and processes documented. Align your solutions with user and business needs while avoiding overengineering.

Design For Resilience

Be prepared for failures. Identify critical component paths, fail-safes, and isolation by implementing segmented solutions and modular design patterns.

- Prioritise critical components and potential failure points that cannot fail without complete disruption.
- Modularise and segment solutions into smaller, independent parts.
- Over-provision critical components to accommodate surges and sudden demand increases.

Resilient solutions ensure seamless user experiences and minimal impact on operations, even during faults.

Design For Recovery

Prioritise a strong DRP. Be ready to recover quickly from failures with minimal disruption. Plan, test, and automate your recovery processes.

- Produce clear disaster recovery plans that undergo continuous testing with recovery metrics and targets.
- Consider real-time replication or log shipping that aligns with restore protocols and targets, ensuring data integrity.
- Automate recovery processes where possible to mitigate delays and downtime.

With a solid, tested DRP, system recoveries and restores can be triggered quickly, minimising business downtime, business impact, and data loss.

Design For Operations

Anticipate issues and faults. Triage logged problems and conduct RCAs so you can adapt and learn from them, driving structured incident management.

- Consider monitoring tools to gain visibility into performance and detect issues early. Alerts help you be preemptive instead of reactive.
- Test faults regularly through simulations and structured ALM frameworks and strategies.
- Automate repetitive tasks, allowing you to reduce mistakes and save time.

Stay ahead of problems by tracking incidents and adapting your response and processes based on lessons learnt.

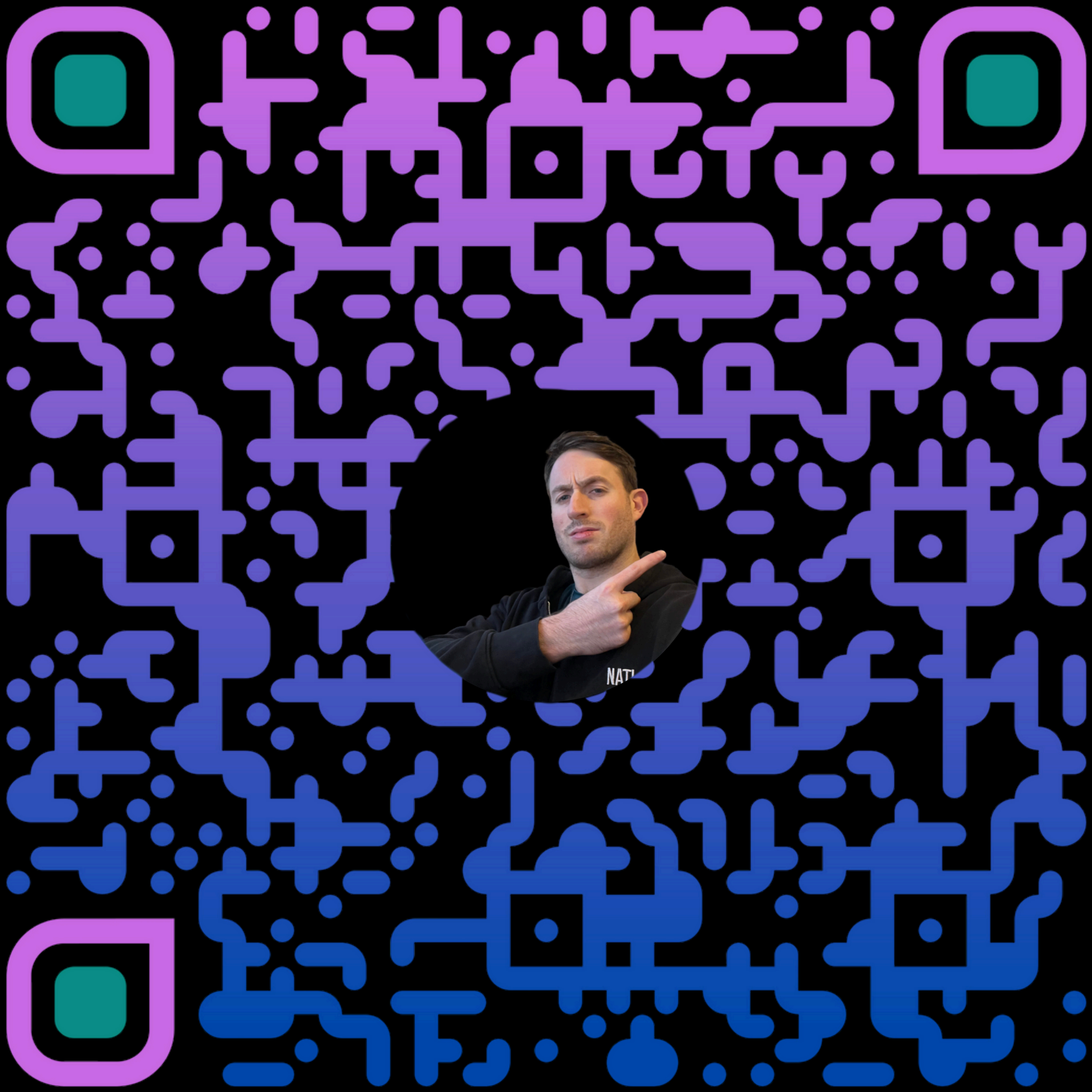
nati-turtledove.com
@NatiTurts

Keep It Simple

A term to become familiar with. Overengineering can distract from business requirements. Focus on what is essential and meets business goals.

- Focus on meeting business requirements for a core solution and deliverable.
- Build along consistent development standards and deployment strategies.
- Use existing platform components and tools where you can.

Simple solutions are easier to maintain and are more reliable, thus resulting in reduced risk.



nati-turtledove.com
@NatiTurts