BLUESKYTEC

# Reliability and Availability Findings for Keyspace Technology

Public Technical Summary

Blueskytec America Inc. | blueskytec.us
Blueskytec Ltd. | blueskytec.com
Version 4.00P | Public Release Version | 15 Jun 2026

This public version of a detailed report submitted 21 Jan 2025, accepted by the U.S. DoW. It summarizes high-level reliability and availability of Keyspace Technology (KST) design. It intentionally omits protected implementation details, including internal product taxonomy, device-level architecture, production methods and restricted implementation details, component identifiers, detailed reliability calculations, failure-path specifics, and deployment-specific configurations.

## SUMMARY

KST is designed to protect machine-to-machine communication in industrial and cyber-physical environments without adding a fragile software dependency at the edge. The reliability finding is simple: KST shifts the availability burden away from complex software execution and toward fixed-function hardware, power design, thermal design, and deployment architecture.

The result is a protection model intended to preserve both integrity and availability. KST does not treat security as useful if it makes the protected system harder to operate, harder to maintain, or more likely to fail. In high-consequence environments, the protection layer must not become the new weak link.

Internal reliability analysis supports the public-facing conclusion that KST has a high-reliability design profile when compared with conventional processor-based edge security approaches. The strongest public finding is not that any single device can never fail. The finding is that KST reduces common edge-security reliability liabilities, including runtime software dependencies, high power demand, thermal stress, active cooling, and persistent external management dependencies.

## PUBLIC FINDINGS

| Finding | Public Description | Operational Meaning |
|---|---|---|
| Fixed-function enforcement | Core enforcement is implemented as a constrained hardware function rather than as a general-purpose software stack. | Fewer runtime states, fewer patch-cycle dependencies, and a smaller operational failure surface. |
| Low-power architecture | The design emphasizes low power consumption and thermal simplicity. | Reduced heat load, reduced cooling demand, and lower reliance on mechanical components. |
| Operational independence | Core protection does not require continuous external management, cloud service availability, or an always-on controller to perform its | Support systems are not required to remain continuously available for core protection to operate. |

| Finding | Public Description | Operational Meaning |
|---|---|---|
| | intended enforcement function. | |
| Power-path priority | Reliability analysis indicates that power delivery deserves the most attention in edge hardware availability planning. | High-availability design should prioritize resilient power, clean installation practice, and appropriate redundancy. |
| Authenticated redundancy | Redundant communication paths are most useful when availability is paired with message integrity. | Failover paths should not merely be reachable. They must also carry trusted, conforming messages. |
| Validation path | Analytical reliability modeling supports design confidence, but it is not a substitute for formal accelerated reliability testing. | Public claims should distinguish design analysis from externally validated field statistics. |

## 1.0 PURPOSE AND SCOPE

This paper provides a public-facing summary of the reliability and availability rationale behind Keyspace Technology design. It is written for customers, partners, investors, and technical audiences that need to understand the findings without receiving protected design information.

The paper does not disclose internal board architecture, device taxonomy, component identifiers, production procedures and restricted implementation details, detailed reliability calculations, or failure-path details. Those topics belong in controlled technical disclosures under the appropriate agreement and release review.

The central question addressed in the original work was not whether hardware can fail. All hardware can fail. The question is whether the protection layer provided by KST introduces avoidable fragility into an environment where availability, safety, and operational continuity matter. KST is designed so that the answer is no.

## 2.0 PUBLIC TECHNICAL BASELINE

KST is a hardware-based integrity and enforcement technology for control and machine-to-machine environments. At a public level, it can be understood as a Deterministic Protection System (DPS) that creates an assured trust layer at or near the execution boundary, where commands, telemetry, or machine messages must be trusted before they affect operations.

KST is not a general-purpose computing platform, software agent, firewall replacement, or cloud-managed appliance. Its public distinction is narrower and more important as KST is designed to enforce trust in machine communications without relying on a conventional runtime software stack for the core protection function.

Core attributes include:

- Hardware-based enforcement of defined machine communication paths (Operational Envelope)
- Low power and reduced thermal burden
- No runtime software dependency or patch management
- No active cooling requirement in standard design assumptions
- No external management dependency for security enforcement
- Support for redundant network, power and control architectures

## 3.0 RELIABILITY DESIGN PRINCIPLES

### 3.1 Deterministic hardware over runtime adaptation

High-availability control environments suffer when the protection layer becomes another complex computing environment. KST avoids that pattern by keeping the core enforcement function constrained and deterministic. The public claim is not that hardware is invincible. It is that a narrowly bounded hardware function can reduce runtime ambiguity compared with a general-purpose software stack.

### 3.2 Reduced software dependency

Conventional edge security systems often depend on operating systems, applications, device drivers, update cycles, restart behavior, certificates, agents, logs, and management services. Those elements can be valuable, but they also expand the operational failure surface. KST is designed so the core protection function does not depend on software to execute trust or security operations.

### 3.3 Power and thermal simplicity

Power and heat are reliability problems disguised as ordinary engineering details. High-power systems require more conversion, more cooling, more mechanical assistance, and more installation discipline. KST is designed around a low-power profile so the supporting physical design can remain simple and more reliable.

### 3.4 Management independence

Support and management channels can become availability dependencies. KST is designed so core enforcement remains local and does not require contact with an external service, centralized controller, or cloud system to perform its intended function.

### 3.5 System-level redundancy

High availability is not achieved by pretending single devices cannot fail. It is achieved by designing the installation, so failure does not become loss of mission function. KST can support redundant deployment models where power, network path, and protected communication design are aligned to the customer environment.

### 3.6 Operational Simplicity and Adaptability

Many conventional security tools require changes to the host environments they are meant to protect. Those changes can introduce new configuration burdens, integration dependencies, maintenance requirements, and potential failure points. KST is designed to reduce that burden. It operates as an independent enforcement layer and does not require changes to the protected device's operating system, application stack, network stack or control logic. This allows KST to support a broad range of operational environments, from serial communications to IP-based networks, across multiple physical media. This adaptability is important in high-availability environments because reliability is not only a function of component durability. It is also a function of how little disruption the protection mechanism introduces into the system it is meant to preserve.

## 4.0 RELIABILITY FINDINGS

The internal reliability analysis found that the dominant availability considerations for a KST-class device are not the deterministic enforcement function itself, but the supporting physical system, especially power delivery and thermal management. Which are more visible, more testable, and more manageable than an uncontrolled software failure surface.

The analysis also supports a broader architectural conclusion that low-power fixed-function hardware can avoid several common reliability penalties associated with processor-heavy edge devices. Those penalties include higher heat generation, reliance on fans or aggressive cooling, greater power-conversion burden, larger software stacks, and more frequent patch or configuration intervention.

These findings support KST as a strong candidate for high-availability environments, provided the installation design, power quality, redundancy model, and lifecycle testing are handled with the same discipline expected of other critical infrastructure components.

| Reliability Area | Public Finding | Engineering Implication |
|---|---|---|
| Core logic | The core enforcement function is constrained and deterministic. | Testability and repeatability improve when the protected function has fewer runtime states. |
| Software chain | The design reduces dependence on operating systems and runtime applications for core enforcement. | Patch cycles, crashes, driver behavior, and software update failure are less central to availability. |
| Thermal profile | Lower power reduces thermal burden. | Reduced heat improves reliability and simplifies installation constraints. |
| Mechanical cooling | The design avoids reliance on active cooling in standard assumptions. | Removal of fans and related mechanical dependencies can improve service life. |
| Power delivery | Power delivery remains the dominant area for reliability engineering attention. | Installation planning should prioritize clean power, resilient supply design, and deployment redundancy. |
| External services | Core protection does not require persistent external management services. | Loss of a management channel does not automatically mean loss of core enforcement. |

## 5.0 HIGH AVAILABILITY MODEL

Traditional high-availability designs often rely on redundant channels, redundant devices, and voting logic. Those patterns remain useful, but cyber-physical environments add a complication in that a path can be available while the message on that path is not trustworthy. As such, availability and access alone does not equate to execution legitimacy.

The finding is that availability and integrity should be coupled. A backup path should not be accepted merely because it is reachable. It should be accepted because the message remains valid, conforming, and trusted under the rules established for the protected environment. This creates a different high-availability posture. In conventional voting models, the system may rely on multiple channels agreeing. In a KST-supported model, the system can prioritize valid, authenticated communication across available paths. The operational principle is direct in that one trusted message is more useful than multiple untrusted votes.

## 6.0 LIMITS OF THE PUBLIC FINDINGS

The public findings should be read with the following limits:

- They are not a warranty that any individual unit will never fail.
- They are not a substitute for customer-specific high-availability engineering.
- They do not disclose, and should not be interpreted to imply disclosure of, protected implementation details.
- They do not replace formal accelerated reliability testing, environmental qualification, or operational field data.
- They do not claim that hardware-based systems are universally superior to software-based systems for every use case.
- The findings are specific to KST’s intended edge enforcement role.

## 7.0 CONCLUSION

The core public conclusion is that Keyspace Technology was designed with reliability and availability as first-order requirements, not as afterthoughts.

Its architecture avoids adding a complex software control point at the edge of high-consequence machine communication, reduces common power and thermal liabilities, and supports redundant deployment models where availability remains tied to message integrity.

The reliability model points to the practical engineering truth beneath the marketing language that high availability is not achieved by wishing devices never fail. It is achieved by reducing

avoidable failure modes, isolating the right dependencies, and designing the protected system so that trust and continuity survive adverse conditions.

For customers and partners, the public takeaway is clear. KST is not merely a security function added to a control environment. It is an availability-conscious enforcement layer designed for environments where operational continuity, message integrity, and consequence management must be handled together.