Acquiring Satellite-Control Software: Source Code, Access and Supply-Chain Diligence

An Evidence-Led M&A Framework for Mission-Control Software, Transferability and Operational Continuity

Authored by Chennakeshav Adya

Independent Researcher

September 2026

Abstract

Satellite-control software can determine whether an acquirer receives a functioning mission capability or an incomplete collection of licences, binaries, interfaces and dependent services. The asset may coordinate planning, command generation, telemetry processing, flight dynamics, ground-station scheduling, anomaly response and customer delivery. Its value depends on executable access, configuration knowledge, controlled source, build reproducibility, specialist people, third-party rights, secure development, vulnerability response and evidence that the software works for the specific fleet and operating model being acquired.

This paper develops an evidence-led M&A framework for acquiring satellite-control software. It translates software engineering, assurance and supply-chain standards into transaction questions, evidence requests, valuation adjustments, closing conditions and post-close controls. The framework distinguishes legal ownership from practical control; source-code possession from buildability; a successful demonstration from repeatable operations; supplier support from transferable rights; and technical debt from mission continuity risk.

The analysis draws on NIST's Secure Software Development Framework and Cybersecurity Supply Chain Risk Management guidance; NASA software engineering and assurance requirements; CISA software-supply-chain and Software Bill of Materials guidance; ESA mission-operations software practice; and space-system protection standards. These sources define useful evidence and control expectations. They do not establish the quality, transferability, security or value of any identified company or software product.

A wholly hypothetical acquisition illustrates the method. The target provides mission-control and flight-dynamics software supporting 18 satellites and 7 ground sites. The seller presents USD 84 million of headline enterprise value. Diligence identifies incomplete build provenance, two non-transferable software dependencies, concentrated administrator knowledge, deferred vulnerability remediation and a customer-specific interface that lacks clean ownership evidence. The illustrative valuation bridge deducts USD 9 million for remediation and transition, USD 7 million for rights and dependency exposure, USD 5 million for concentration and operational fragility, and USD 4 million for contingent customer acceptance. A conditional USD 6 million earn-out can be released against verified build reproducibility, licence novation, access transfer and service continuity. The resulting illustrative cash value at closing is USD 59 million.

The central conclusion is that satellite-control software should be acquired through an evidence chain. A buyer should be able to identify what runs, reproduce how it is built, prove who can operate it, establish who owns and may transfer each component, test recovery, and connect unresolved dependencies to price and closing mechanics. Value follows demonstrable control over the software system and its mission outcomes.

JEL Classification: G34, L63, L86, L96, M15, O32, O33

Keywords: satellite control software, source code diligence, software supply chain, space M&A, mission operations, intellectual property, software escrow, cybersecurity, operational continuity, valuation

Introduction

Software sits between the satellite, the ground network, operators, customers and regulators. Mission-control systems plan contacts, validate and transmit commands, receive and process telemetry, monitor spacecraft health, calculate orbits, manage payload schedules and preserve the operational record. A defect, unavailable dependency or inaccessible signing key can interrupt revenue and weaken command authority even when the underlying spacecraft remains technically healthy.

An acquisition therefore involves more than a conventional software-product review. The buyer must understand the mission system as operated. That system includes source repositories, build pipelines, configuration data, deployment environments, cryptographic material, ground interfaces, runbooks, supplier services, engineering judgement and contractual rights. Several of these elements may sit outside the target's legal entity or depend on named individuals.

This paper sets out a transaction framework for that problem. It is designed for strategic acquirers, infrastructure investors, private-capital firms, satellite operators, lenders and boards assessing a software-led space transaction. It focuses on evidence that changes control, continuity, price, conditions and integration design.

1 Define the acquired capability

The buyer should begin with a capability statement. The statement identifies which missions, spacecraft, payloads, ground sites, customer services and operational decisions the software supports. It records service windows, availability expectations, safety consequences, regulatory obligations and revenue dependencies. Product names alone provide an unreliable perimeter because one branded platform can depend on separate flight-dynamics, planning, identity, database, monitoring and ground-station services.

The capability statement should distinguish flight software from ground software. Satellite-control acquisitions usually focus on the ground segment, while operational value can depend on embedded flight interfaces and spacecraft-specific command databases. The buyer needs evidence that the target may use, modify and transfer every interface needed for continued operations.

The perimeter should also identify excluded services. Shared corporate identity, cloud subscriptions, communications links, key-management services, data centres or parent-company staff may be essential on day one. Each exclusion becomes a transition requirement, continuing dependency or value adjustment.

2 Separate ownership, access and control

Legal ownership is one element of control. The buyer also needs physical or logical access, sufficient rights to modify and deploy, knowledge to operate, and authority over credentials and release decisions. These elements can diverge. A target may own custom source code while relying on a non-transferable library. It may possess a repository while the production build depends on a consultant's private toolchain. It may hold broad licences while a government customer controls deployment approval.

The diligence model should record ownership, possession, access, modification rights, distribution rights, operational authority and termination rights separately. Each item needs documentary evidence. Relevant evidence includes assignments, employment terms, contractor agreements, licence schedules, repository permissions, deployment records, customer contracts and supplier confirmations.

Control also has a time dimension. Access that exists during diligence may expire at closing. The buyer should identify the exact actions required to preserve repository access, cloud tenancy, signing authority, administrator credentials, monitoring history and supplier support through the transaction boundary.

3 Build the software and dependency inventory

The inventory should identify applications, services, repositories, branches, build systems, deployment packages, databases, interfaces, commercial products, open-source components, cryptographic libraries and operational scripts. It should connect each component to the mission function it supports and the environment in which it runs.

An SBOM can accelerate component discovery. It does not replace the operating inventory. A useful acquisition inventory links component name and version to source, licence, maintainer, vulnerability status, build artefact, deployed configuration, data dependency and replacement path. Components embedded in containers, firmware or vendor appliances require attention because they can be absent from a conventional application list.

The buyer should reconcile three views: what engineering says exists, what repositories contain and what production telemetry shows running. Differences are diligence findings. An unrecorded service may be critical. A listed repository may be obsolete. A production binary may not trace to an approved source commit.

4 Test source-code completeness

Repository access should cover the full product and its history. The buyer should inspect source, configuration, infrastructure definitions, database schemas, test assets, build scripts, deployment automation, documentation and issue history. A snapshot of selected files cannot demonstrate completeness.

Completeness testing starts from deployed artefacts. The team identifies representative production binaries or containers and traces them back to source revisions, dependency versions, build parameters and approval records. It then builds the software in a controlled environment and compares the result with the deployed artefact or its documented provenance. Differences require explanation.

Generated code deserves separate treatment. NASA guidance recognises that long-term maintenance can require access to models, simulations, data definitions, generators, build data, test scripts and expected results. Possession of generated source can be insufficient when the generator, model or qualified configuration is missing.

5 Reproduce the build

A reproducible build is a high-value diligence test. The objective is to show that an authorised team can create a deployable release from controlled inputs without undocumented intervention. The test should use a clean environment and the target's documented procedure. Buyer observers should record prerequisites, external downloads, credentials, manual steps, tool versions, warnings and deviations.

The result need not be bit-for-bit identical when the existing process does not support deterministic compilation. It should be traceable and functionally equivalent under agreed tests. The buyer should understand why any difference arises and whether the process preserves integrity.

Build failure can reveal missing licences, expired certificates, unavailable package repositories, undocumented patches or reliance on a named engineer. These findings connect directly to transition cost and continuity risk. The purchase agreement can require a successful clean build before closing or place consideration in escrow until the condition is met.

6 Trace release and deployment authority

Source access does not establish the ability to operate production. The buyer should trace authority from code approval through build, signing, release, deployment and rollback. The trace identifies who can approve changes, who holds signing credentials, where packages are stored, how environments are promoted and how emergency releases are controlled.

Satellite-control changes can have mission-safety consequences. Release evidence should include verification results, operational readiness review, configuration approval, customer or authority consent where applicable, and a tested recovery route. The buyer should distinguish routine application releases from changes to command validation, flight dynamics, telemetry interpretation or cryptographic trust.

Credential transfer requires a designed process. Private keys and privileged credentials should not be copied casually during closing. The parties should agree rotation, revocation, re-enrolment, dual control, audit preservation and rollback. The closing plan should state when operational authority changes and how ambiguous responsibility is avoided.

7 Examine mission configuration data

Mission-control software relies on configuration that can be as valuable as source code. Command dictionaries, telemetry definitions, limits, calibration data, spacecraft models, contact plans, orbital parameters, automation rules and customer routing determine how generic software interacts with a real fleet.

The buyer should identify the authoritative source for each configuration class, its approval process, version history and recovery method. It should test whether a new environment can be populated from controlled records. Spreadsheet-based or locally stored configuration creates risk when it lacks review, lineage and backup.

Configuration rights also matter. A customer, spacecraft manufacturer or systems integrator may own or restrict part of the data. The acquisition agreement should address transfer, continuing use, confidentiality, export restrictions and deletion obligations. Missing configuration can make otherwise complete software unusable for a specific mission.

8 Assess software assurance evidence

Software assurance evidence shows whether the product was developed and maintained through controlled practices. NIST's SSDF organises secure development around preparing the organisation, protecting software, producing well-secured software and responding to vulnerabilities. NASA software-assurance guidance adds objective evidence for safety and mission contexts.

The buyer should review requirements traceability, architecture decisions, code review, static analysis, dependency scanning, test coverage, release approval, defect history and vulnerability response. Evidence should correspond to the versions in production. A policy document without executed records provides limited assurance.

The diligence team should sample critical paths such as command generation, authentication, privilege management, orbit determination and recovery. It should examine whether tests cover adverse and boundary conditions. Open findings should be classified by mission consequence, exploitability, recoverability and remediation dependency.

9 Map the software supply chain

The supply chain includes commercial vendors, open-source projects, cloud providers, build services, package repositories, hardware suppliers, consultants and specialist operators. The buyer should map which parties can change, interrupt, access or constrain the product.

For each critical supplier, diligence should cover contractual support, financial resilience, security practices, access privileges, incident notification, change control, end-of-life policy, data location and substitution lead time. A supplier that supports several mission-critical components creates concentration even when annual spend is small.

NIST SP 800-161 treats supply-chain risk as an organisational governance problem rather than a procurement checklist. The transaction team should connect supplier evidence to system criticality and planned ownership. Material suppliers may require consent, novation or direct agreement before closing.

10 Analyse open-source exposure

Open-source components can improve capability and reduce development time. They also create licence, maintenance and security obligations. The buyer should reconcile declared components with code scanning and build manifests. It should identify licence terms, modifications, notices, distribution practices and known vulnerabilities.

Copyleft exposure requires legal analysis based on actual use and distribution. A licence label alone does not establish the obligation. The team should document how components are linked, deployed, modified and provided to customers. Remediation can involve notice correction, source offers, component replacement or customer communication.

Maintenance status affects value. A critical library with no active maintainer or supported release can require internal ownership. The buyer should assess fork strategy, test coverage, patch capability and community dependence. A software bill of materials should remain current after closing rather than becoming a static diligence artefact.

11 Review intellectual-property provenance

Every material code contribution should have a defensible provenance path. Employee inventions should fall within appropriate employment terms. Contractor work should be assigned with sufficient scope. Acquired or contributed code should carry the necessary rights. University, government or customer-funded development can include restrictions that require detailed review.

The team should sample commit history against contributor records and contracts. Unrecognised authors, personal accounts or unexplained bulk imports deserve investigation. The review should include documentation, models, test data, user interfaces and algorithms, because valuable intellectual property extends beyond source files.

Patent and trade-secret analysis should focus on the actual product and commercial plan. The buyer should understand defensive registrations, freedom-to-operate work, confidentiality controls and any disclosure that could weaken trade-secret protection. Transaction documents can allocate specific provenance risks through warranties, indemnities, escrows or contingent consideration.

12 Test privileged access and segregation

Mission-control environments concentrate powerful privileges. Administrators may control commands, identity, databases, cloud resources, signing keys and monitoring. The buyer should obtain a privilege inventory that covers human, service and emergency accounts across development, test, production and recovery environments.

The review should test joiner, mover and leaver controls; multifactor authentication; approval; session logging; credential rotation; break-glass access; and segregation between development and production. Shared or dormant accounts reduce accountability. Service accounts require ownership and lifecycle controls.

Closing creates heightened access risk. Departing personnel may retain credentials or knowledge of recovery paths. The transition plan should rotate sensitive credentials, revoke old access, preserve evidence and maintain dual operational coverage. These actions should be rehearsed where loss of access could affect a live mission.

13 Review vulnerability management

The buyer should understand how vulnerabilities are discovered, triaged, remediated and communicated. Sources include static analysis, dependency scanning, penetration testing, customer reports, supplier advisories and operational monitoring. The programme should define severity, mission consequence, remediation timing, exception approval and retesting.

Backlog analysis can reveal hidden maintenance requirements. The buyer should examine age, recurrence, affected versions and closure quality. A low count can reflect weak detection. A high count can reflect active discovery. The useful measure is the organisation's ability to identify relevant exposure and reduce it within risk-based timeframes.

Flight and ground systems may have limited patch windows. The target should show how it applies compensating controls and plans safe releases. Known vulnerabilities in inaccessible or end-of-life components can justify a specific price deduction or funded remediation reserve.

14 Assess interfaces and interoperability

Satellite-control software depends on interfaces with spacecraft, ground stations, networks, identity systems, customer platforms, weather data, orbit data and regulatory services. The buyer should inventory protocol, version, owner, performance requirement, security control, test environment and change process for each interface.

Interface documentation should be tested against observed traffic and current configurations. Proprietary or undocumented interfaces create lock-in. A standard protocol can still contain customer-specific extensions that complicate replacement.

The team should test failure modes. It should observe delayed data, duplicate messages, clock errors, corrupted input, network interruption and partial service loss. Recovery behaviour affects operational value. Integration plans should preserve interface observability and avoid changing several critical dependencies at once.

15 Evaluate data rights and records

Operational data supports monitoring, anomaly analysis, model improvement, customer reporting and regulatory evidence. The buyer should distinguish ownership, custody, permitted use, retention and export rights for telemetry, commands, derived products, logs, customer information and training data.

The transaction perimeter should include historical data needed to operate and improve the system. A buyer may receive software without sufficient operational history to tune alerts or investigate anomalies. Data migration should preserve timestamps, provenance, access controls and evidential integrity.

Privacy and localisation obligations can apply to personnel and customer data. Export controls and national-security restrictions can affect technical data and access by foreign persons. The legal analysis should map obligations to actual datasets and operating locations.

16 Examine operational resilience and recovery

Resilience testing should show how the service continues through infrastructure, software, supplier and human failures. The buyer should review redundancy, backups, recovery environments, communications alternatives, manual procedures, recovery objectives and exercise results.

A backup is useful only when it can be restored within the mission's tolerance. The diligence team should observe a representative restore and validate configuration, credentials, dependencies and data integrity. Recovery should include the ability to rebuild systems when the primary environment or supplier is unavailable.

The acquisition itself should be treated as a resilience event. Corporate systems, domains, cloud accounts and support contracts may change. A detailed cutover plan, rollback criteria, command authority and joint incident process reduce transition risk.

17 Assess people and knowledge concentration

Software continuity depends on people who understand architecture, missions, anomalies, customers and operational judgement. The buyer should map roles to critical processes and identify single points of knowledge. Organisational charts provide limited evidence; the map should reflect who actually resolves incidents and approves releases.

Retention decisions should consider both technical importance and independence. Concentrated knowledge can be reduced through pairing, documentation, rehearsal and succession. Retention payments without knowledge-transfer milestones can preserve dependence rather than reduce it.

The transaction model should include recruitment, retention, contractor conversion and training costs. Key-person risk can affect valuation when capability cannot be transferred within the transition period. Management should report progress against named knowledge-transfer evidence.

18 Review customer and regulatory acceptance

Customer contracts can define software performance, security, audit, approval and change obligations. The buyer should identify contracts that require consent, notice, recertification or named personnel. It should separate revenue that transfers automatically from revenue dependent on customer acceptance.

Government and critical-infrastructure customers may impose technical-data, security-clearance, hosting or supply-chain restrictions. The buyer should assess whether its ownership, financing, personnel and operating model remain eligible. Regulatory licences and spectrum rights may sit outside the software entity while remaining essential to service delivery.

The commercial model should probability-weight uncertain renewals and consents. Consideration can be contingent on verified transfer or retained revenue. The buyer should avoid paying full value for contracts whose operating prerequisites do not transfer.

19 Connect diligence to valuation

Software diligence changes value through cash flow, timing, risk and option life. Remediation increases cost. Missing rights can reduce addressable revenue. operational fragility can increase outage probability. Weak build control can delay product development. Strong evidence can support lower integration reserves and greater confidence in renewal.

The valuation bridge should avoid duplicating adjustments. A recurring support cost belongs in forecast cash flow. A one-time rebuild belongs in transaction or integration uses. A binary rights defect may require exclusion, escrow or a condition rather than a discount rate premium.

The buyer should show base, downside and severe cases. Each case should identify operational assumptions, evidence and management actions. Terminal value should reflect ongoing maintenance, component obsolescence, supplier concentration and the ability to refresh the product.

20 Translate findings into transaction mechanics

Material findings should have an owner and a transaction response. Possible responses include price reduction, holdback, escrow, earn-out, indemnity, closing condition, covenant, transition service, licence novation, key-person arrangement or excluded asset.

Conditions should be objectively testable. A requirement to provide complete source is weak without defined repositories, branches, artefacts and acceptance tests. A build condition can specify environment, inputs, successful test suite and deployment package. An access condition can specify identities, privileges, keys and verified login.

The disclosure process should preserve versioned evidence. The buyer should record which artefacts support each representation and who accepted exceptions. Technical schedules need review by engineering and counsel so that legal language corresponds to operational reality.

21 Design the closing control room

Closing should be managed as an operational change. The control room coordinates legal completion, credential transfer, supplier novation, cloud tenancy, code repositories, signing authority, customer notices, monitoring and incident coverage.

The plan should use explicit go, hold and rollback criteria. Each step identifies evidence, timing, accountable person and fallback. Critical access should be tested before irreversible actions occur. The parties should maintain joint coverage during the highest-risk window.

The buyer should preserve logs and configuration snapshots. These records establish the state at transfer and support later investigation. The first post-close release should follow the agreed assurance process rather than becoming an improvised integration test.

22 Establish the first 100 days

The first 100 days should stabilise control, close priority evidence gaps and reduce concentration. Early actions include access recertification, credential rotation, clean builds, dependency confirmation, critical backlog remediation, recovery testing, staff retention and supplier governance.

Architecture changes should follow evidence. An immediate platform migration can combine ownership transition with technical change and create avoidable risk. The integration team should sequence changes around mission windows, customer commitments and recovery capacity.

The board should receive a concise dashboard covering build reproducibility, critical vulnerabilities, key access, supplier novations, customer consents, recovery tests, knowledge transfer and spend. Reported completion should require objective evidence.

23 Govern continuing software value

Post-close governance should connect software health to commercial and financial outcomes. Engineering measures include deployment reliability, defect escape, vulnerability age, build integrity, recovery performance and dependency status. Commercial measures include service availability, renewal, delivery latency and support cost.

The product roadmap should distinguish mandatory maintenance, customer commitments, security remediation and growth investment. Deferred maintenance can inflate short-term earnings while weakening future cash generation. Investment committees should understand that distinction.

The buyer should maintain an evidence register for critical rights, configurations and suppliers. Changes in ownership, licence, support or architecture can alter the acquisition thesis. Annual assurance and periodic transaction-readiness reviews preserve optionality for refinancing or exit.

24 Hypothetical transaction case

The hypothetical target supports 18 satellites across communications, earth observation and hosted-payload missions. Its platform includes mission planning, command validation, telemetry processing, flight dynamics, automation and customer delivery. Revenue is generated through annual software and operations contracts.

Diligence confirms strong customer retention and capable engineering. It also identifies five transaction exposures. Production builds require an undocumented commercial tool. Two libraries are licensed to the seller's parent and cannot transfer automatically. One administrator controls key production and signing processes. Critical vulnerability remediation has been deferred around mission windows. A customer-specific adapter contains contributions with incomplete assignment evidence.

The buyer prices these findings through a USD 25 million aggregate deduction from the USD 84 million headline value. A further USD 6 million earn-out is available when defined evidence gates are met. The structure preserves seller participation in remediation while protecting buyer cash at closing.

25 Decision principles

First, define the acquired mission capability and its operational perimeter. Second, trace every critical deployed artefact to controlled source, build and approval evidence. Third, separate ownership, access, rights and operational authority. Fourth, map supplier and people concentration to continuity. Fifth, test recovery and transaction cutover. Sixth, translate each unresolved dependency into cash, condition or contractual protection.

These principles create a common language for engineering, finance, operations and legal teams. They also improve transaction speed because evidence requests and acceptance tests become specific.

The buyer's objective is demonstrable control of mission outcomes. That control supports continuity, customer confidence, integration and defensible valuation.

26 Review architecture and technical debt

Architecture diligence should explain how the system separates mission planning, flight dynamics, command validation, telemetry, automation, identity, data stores and external interfaces. The buyer should identify trust boundaries, failure domains, shared services and components whose change could affect several missions. Architecture diagrams should be reconciled with deployed topology, network flows and repository ownership.

Technical debt should be expressed through consequences. An old programming language can remain manageable when expertise, tests and toolchains are available. A modern service can be fragile when it lacks ownership, observability or recovery. The diligence team should estimate the cost, sequencing and operational risk of each material debt item.

Debt classification should distinguish maintainability, security, performance, scalability, obsolescence and compliance. The commercial model should reflect the categories that constrain growth or customer

retention. Remediation plans should identify dependencies and mission windows rather than present an undifferentiated backlog.

27 Evaluate observability and incident evidence

Mission-control value depends on the ability to detect and explain abnormal behaviour. The buyer should review logs, metrics, traces, alerts, dashboards, retention, clock synchronisation and incident records. It should identify where data is generated, who can access it and whether records survive a system or supplier failure.

Alert quality deserves sampling. High volumes of unactionable alerts can conceal material events. Sparse alerts can reflect limited coverage. The team should compare selected incidents with recorded telemetry, response actions, root-cause analysis and corrective work. Repeated incidents without durable remediation indicate control weakness.

Transaction planning should preserve monitoring continuity. Account migration, cloud separation or tool replacement can create blind periods. The closing plan should define which alerts remain active, who receives them and how the joint team handles an event during transition. Evidence of stable monitoring can support a shorter transition-services period.

28 Test scalability and fleet expansion

Historical performance does not establish that the platform can support the buyer's planned fleet. The team should identify scale drivers such as satellites, contacts, telemetry volume, command load, customer interfaces, concurrent operators and ground sites. It should review observed peak utilisation and capacity margins.

Load tests should use representative message patterns and failure conditions. Space operations can create short periods of intense activity around launch, commissioning, anomalies and contact windows. Average utilisation can hide queueing, database contention or operator overload during these periods.

The growth case should include infrastructure, licence, support and staffing costs. A platform that scales technically may face commercial limits from per-satellite vendor pricing or scarce specialists. The valuation model should align growth revenue with the capital and operating cost required to deliver it.

29 Assess artificial-intelligence features carefully

Targets may use machine learning for anomaly detection, scheduling, image processing, predictive maintenance or operator assistance. Diligence should identify the exact decision supported, model owner, training data rights, validation, monitoring, human oversight and failure response. Marketing descriptions provide limited evidence of operational value.

The buyer should compare claimed performance with a defined baseline and representative mission data. It should examine false positives, false negatives, drift, retraining, version control and edge cases. A model that improves average detection can still be unsuitable for command decisions when its failure modes are opaque.

Generative tools used in engineering or operations need governance over sensitive data, code provenance and output review. The product roadmap should separate demonstrated customer value from experimental capability. Valuation should recognise AI-related revenue only when rights, performance, deployment and cash conversion are evidenced.

30 Review export control and sovereign-access constraints

Satellite software, technical data and services can be subject to export controls, sanctions, security classifications and sovereign-access requirements. The buyer should map controlled items, licences, nationalities, locations, cloud regions and customer restrictions. Specialist counsel should interpret the applicable legal regime.

Operational access can be constrained even when ownership transfers. Certain customers may require cleared personnel, domestic hosting or segregated support. The buyer's ownership and financing structure can trigger review or consent. These factors affect integration design and the ability to centralise operations.

The transaction model should include compliance staffing, segregated environments, licence timing and possible revenue restrictions. Closing conditions should address approvals that are necessary for continuity. Management should avoid broad assurances and identify the precise scope covered by each authorisation.

31 Create an investment-committee evidence pack

The investment committee needs a concise connection between technical findings and economic decisions. The pack should begin with the acquired capability, revenue dependency and critical control points. It should present the source and build evidence, rights position, supplier map, operational resilience, customer consents and people concentration.

Each material finding should show evidence status, cash consequence, proposed mechanic, accountable owner and residual risk. The committee should be able to distinguish a verified gap from a management estimate and a future remediation promise. Scenario analysis should show the effect of delays and combined failures.

Approval should state conditions, delegated authorities and post-close reporting. The evidence pack then becomes the baseline for integration governance. This continuity reduces the risk that diligence findings disappear after closing and supports later accountability for value creation.

32 Plan separation from the seller

A carve-out requires a separation model covering applications, infrastructure, identities, networks, contracts, data and people. The buyer should identify services that remain with the seller, services that transfer and services that must be rebuilt. Each line should have an interim operating method, target state, cost, dependency and exit criterion.

Transition-service agreements should describe measurable services and operational responsibilities. They should identify service levels, security obligations, incident coordination, access, change control, data return, audit rights, pricing and termination. A broad commitment to provide reasonable assistance can leave critical mission needs unresolved.

The separation schedule should follow mission constraints. Identity or network changes can affect monitoring and command paths. Data migration can affect history and audit evidence. Supplier novation can change support rights. The parties should stage high-risk changes and preserve rollback until acceptance criteria are met.

Exit readiness should be tested before a transition service ends. The buyer should demonstrate independent access, build, deployment, monitoring, support and recovery. It should also confirm that seller accounts and data can be removed without disrupting service. Any extension should have a defined reason, price and remediation owner.

Separation cost belongs in the transaction model. It includes duplicate infrastructure, temporary licences, migration engineering, security testing, customer validation and operating overlap. The estimate should distinguish committed quotations from management assumptions. A funded and governed separation plan protects continuity and prevents the transaction from carrying an indefinite dependency on the seller.

The board should receive a weekly separation dashboard until all mission-critical services operate independently. The dashboard should identify overdue dependencies, tested exits, unresolved customer obligations, forecast cost and the next irreversible action. Closure should require evidence from engineering, operations, security, finance and the relevant service owner.

Appendix A Diligence evidence register

The evidence register should identify the question, requested artefact, source owner, version, review result, exception, financial consequence and transaction response. It should remain linked to the virtual data room so that conclusions can be reproduced.

High-priority evidence includes the production inventory, repository map, clean-build result, deployment trace, privilege inventory, licence schedule, SBOM, vulnerability backlog, recovery test, customer consent map and key-person plan. Each item should identify the exact systems and configurations covered.

Sampling should be risk based. The team should choose representative high-consequence missions, critical interfaces, recent releases, emergency changes and older components. Samples should test both control design and actual execution.

Appendix B Valuation method

The illustrative method begins with enterprise value derived from the buyer's commercial model. It then adjusts forecast cash flow for recurring maintenance and supplier costs. One-time remediation, separation and transition costs are included in uses. Revenue subject to customer consent is probability weighted. Rights defects and operational conditions are addressed through deductions, escrows or contingent consideration.

Scenario analysis should combine related risks. A supplier delay may also delay remediation and customer acceptance. A key-person departure may weaken both continuity and knowledge transfer. Correlated downside deserves explicit treatment.

No hypothetical number in this paper represents an identified transaction. The case demonstrates how evidence can be connected to valuation and transaction structure.

Appendix C Closing acceptance tests

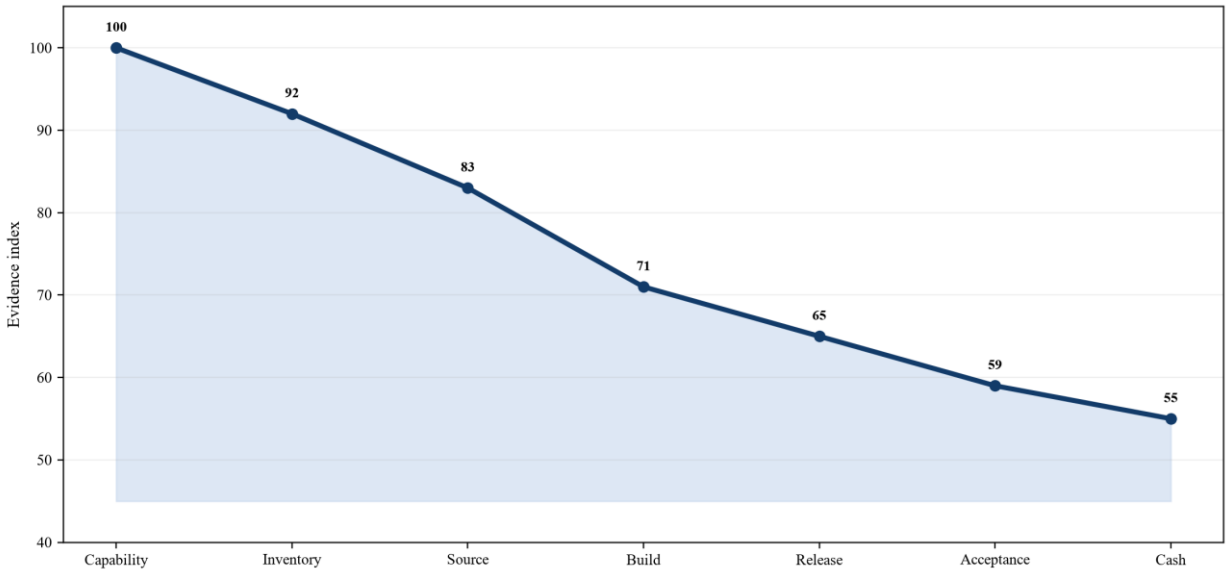
Acceptance tests should cover repository access, clean build, test execution, package signing, non-production deployment, configuration restore, monitoring, privileged access, supplier support and recovery. The parties should agree test data, environment, pass criteria and evidence before signing.

Tests should avoid live operational disruption. Production changes require mission approval and separate controls. A non-production test can still validate the chain from controlled source to deployable artefact.

Failed tests should have predefined consequences. These can include cure, escrow, delayed closing, reduced consideration or exclusion of an asset. The response should match the operational and financial significance of the failure.

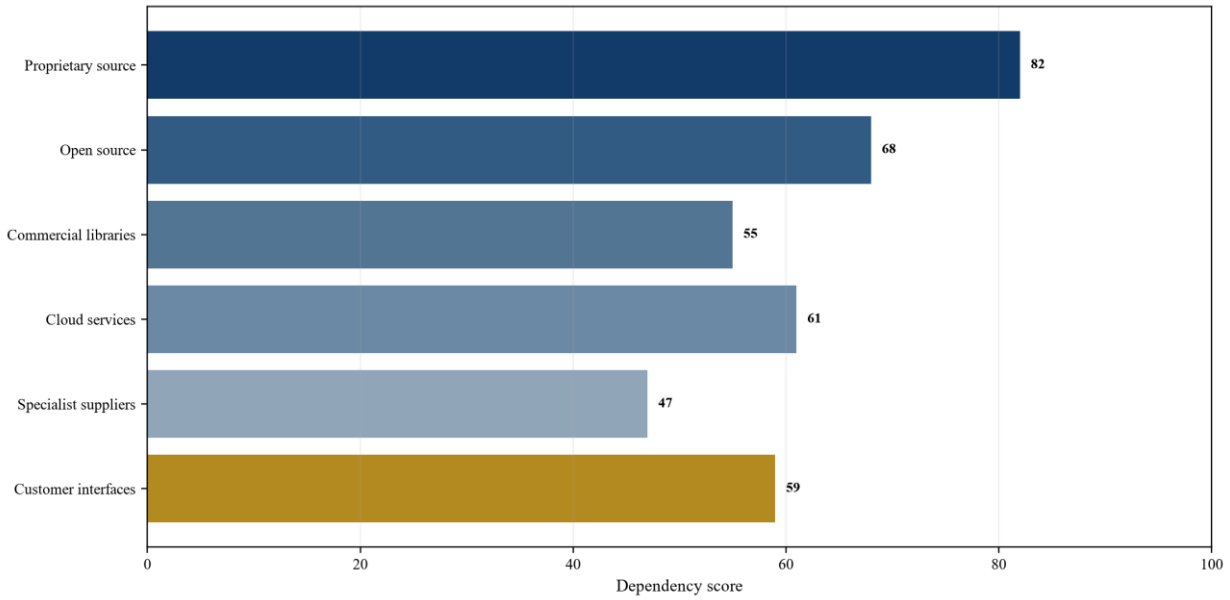
Appendix D Decision figures and tables

Figure 1. Satellite-control software evidence chain



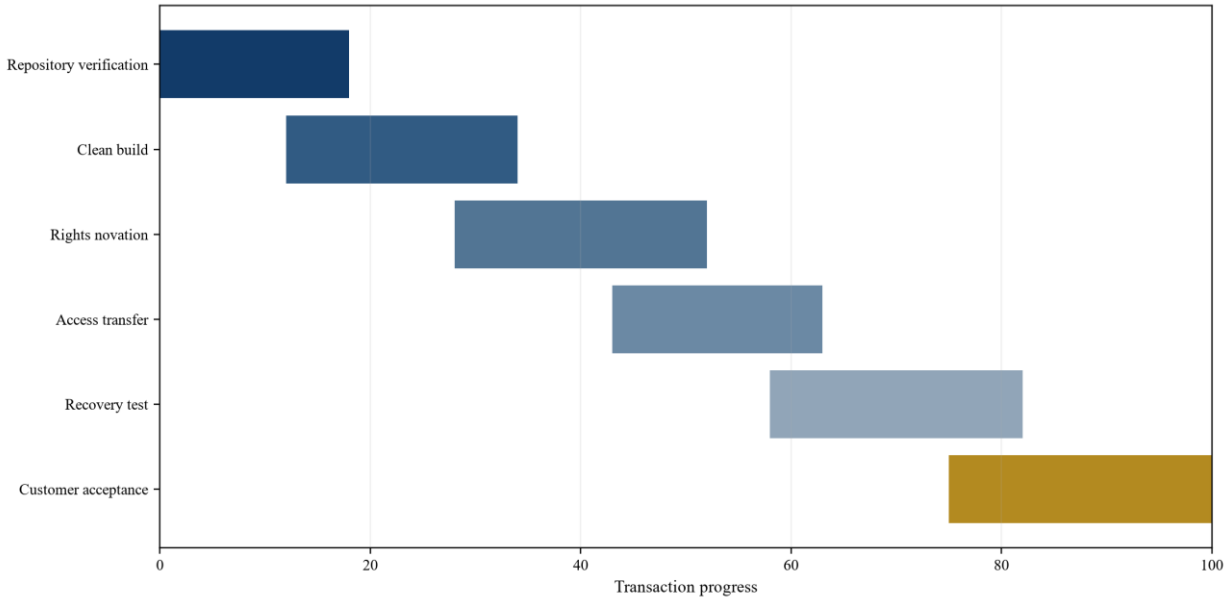
Illustrative evidence progression; values are hypothetical and do not represent an identified company.

Figure 2. Hypothetical source and supply-chain dependency map



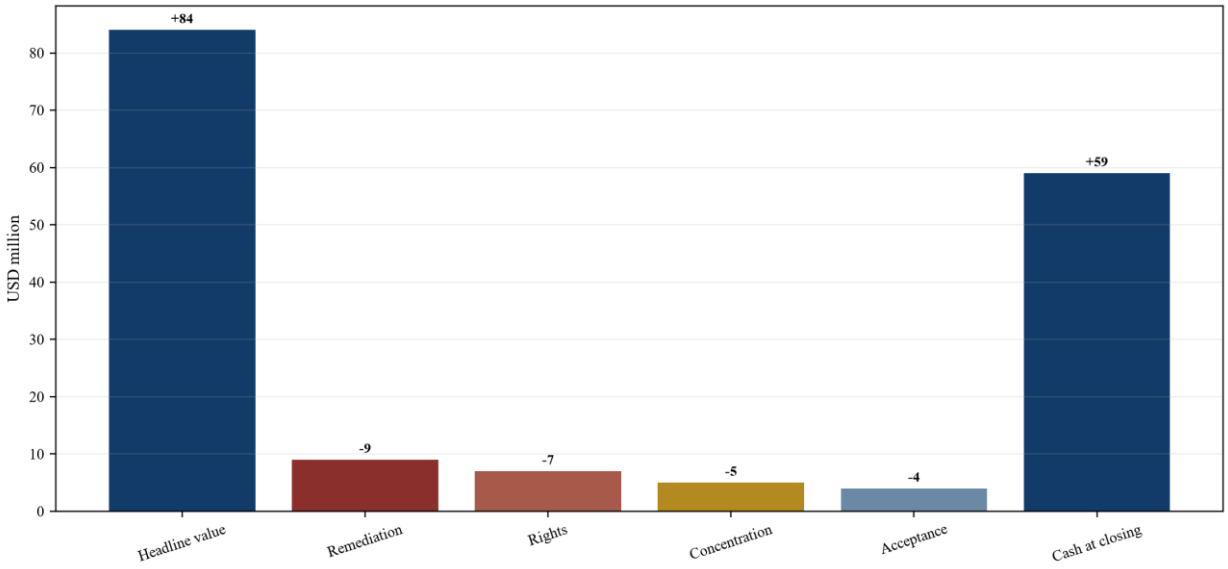
Higher scores indicate greater dependence in the hypothetical case.

Figure 3. Illustrative closing evidence gates



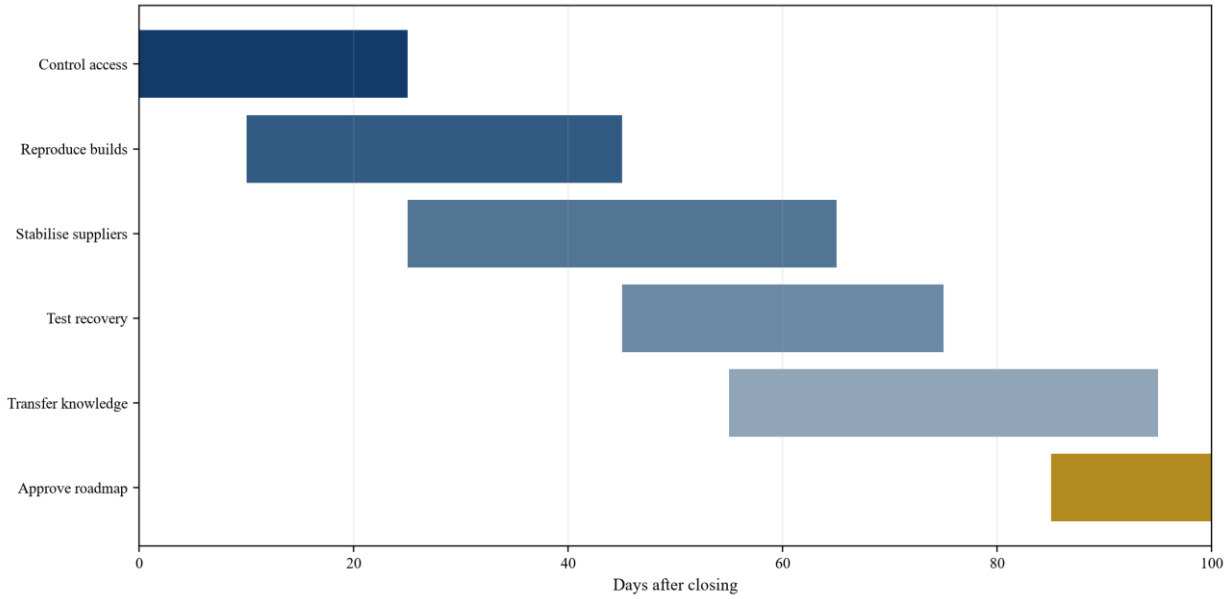
Proposed sequence; actual timing depends on transaction facts and mission constraints.

Figure 4. Hypothetical enterprise-value bridge



Wholly hypothetical values; USD million.

Figure 5. Illustrative first 100 days



Proposed sequence subject to mission windows and customer obligations.

Table 1. Source and build evidence

Evidence	Transaction question	Acceptance indicator	Response to gap
Repository inventory	Is all material source controlled?	Production components trace to named repositories	Completion condition or scoped exclusion
Build definition	Can a clean environment produce a release?	Successful controlled build with documented inputs	Holdback and remediation plan
Provenance	Can deployed artefacts be traced?	Version, dependencies, approval and signature linked	Risk reserve and control remediation
Generated artefacts	Are models and generators available?	Rebuild possible from controlled sources	Licence or asset transfer condition

Proposed transaction evidence requirements.

Table 2. Rights and dependency analysis

Dependency	Evidence	Principal exposure	Transaction treatment
Employee code	Employment and invention terms	Ownership gap	Assignment and warranty
Contractor work	Statement of work and assignment	Restricted modification or transfer	Specific assignment or indemnity
Commercial software	Licence and support terms	Non-transfer, termination or price reset	Novation, replacement or deduction
Open-source software	SBOM, licence and distribution record	Compliance and maintenance	Cure plan and continuing governance
Customer interface	Contract and contribution history	Use restricted to existing customer	Consent or contingent value

Proposed legal and operational dependency classification.

Table 3. Hypothetical valuation bridge

Item	USD million	Evidence basis	Treatment
Headline enterprise value	84	Commercial model	Starting value
Remediation and transition	-9	Build, vulnerability and separation plan	Closing deduction
Rights and dependencies	-7	Licence and provenance gaps	Deduction or escrow
Concentration and fragility	-5	Access and key-person review	Integration reserve
Customer acceptance	-4	Consent and interface evidence	Contingent consideration
Cash value at closing	59	Aggregate framework	Illustrative result

Wholly hypothetical; USD million.

Table 4. Closing control plan

Control	Evidence before transfer	Closing action	Post-close verification
Repositories	Access list and protected branches	Transfer administrators	Reconcile access and logs
Signing	Key inventory and approval chain	Rotate or re-enrol keys	Test signed non-production release
Production access	Privilege inventory	Revoke seller-only accounts	Recertify human and service access
Suppliers	Contract and consent schedule	Execute novations	Confirm support and incident contacts
Recovery	Latest exercise and backup inventory	Preserve snapshots	Run controlled restore

Proposed controls; actual sequencing depends on mission constraints.

Table 5. First 100-day dashboard

Measure	Day 30 evidence	Day 60 evidence	Day 100 evidence
Build control	Clean environment established	Critical products reproduced	Routine release evidence complete
Access	Administrators recertified	Service accounts assigned	Exceptions closed or accepted
Vulnerabilities	Critical backlog validated	Priority cures tested	Sustainable service levels operating
Suppliers	Critical dependencies confirmed	Novations and substitutes progressed	Governance and exit plans approved
Knowledge	Key roles retained	Runbooks and pairing underway	Independent operational coverage tested

Proposed evidence milestones.

Table 6. Risk-to-mechanic mapping

Finding	Cash effect	Contract mechanic	Evidence for release
Non-transferable dependency	USD 7 million	Escrow or price deduction	Executed novation or qualified replacement

Finding	Cash effect	Contract mechanic	Evidence for release
Build fragility	USD 4 million	Closing condition	Successful clean build and test
Key-person dependence	USD 3 million	Retention-linked holdback	Knowledge-transfer milestones
Customer-specific rights	USD 4 million	Earn-out	Consent and retained cash collection

Wholly hypothetical cash effects; USD million.

Table 7. Board decision dashboard

Decision area	Green evidence	Amber condition	Red condition
Source control	Complete traceable repositories	Minor controlled gaps	Missing critical source or history
Build	Repeatable controlled release	Manual steps with funded cure	Build cannot be reproduced
Rights	Transferable documented rights	Consent pending with protection	Material right unavailable
Operations	Independent staffed coverage	Concentration with tested transition	Named-person dependency without fallback
Recovery	Recent successful restore	Partial scope or aged exercise	Recovery untested or unavailable

Proposed decision thresholds.

Frequently asked questions

Q: Does possession of source code prove that the buyer controls the software? A: No. Practical control also requires buildability, deployment authority, configuration data, credentials, operating knowledge, transferable rights and access to dependencies. Each element should be evidenced separately.

Q: What is the most useful technical diligence test? A: A clean, observed build from controlled source is highly informative because it tests completeness, toolchain, dependencies, documentation and staff knowledge. It should be combined with deployment and recovery evidence.

Q: Is a software bill of materials sufficient for supply-chain diligence? A: An SBOM supports component discovery. Acquisition diligence also requires licence, provenance, vulnerability, support, criticality, deployed configuration, supplier access and replacement-path evidence.

Q: When should a source-code escrow be used? A: Escrow can support continuity when a critical supplier retains ownership. The buyer should test deposit completeness, update frequency, release triggers, buildability and rights after release. An untested archive offers limited protection.

Q: How should customer-specific software be valued? A: Value should reflect ownership, transferability, continuing customer rights, maintenance cost and renewal probability. Uncertain consent or assignment can be addressed through contingent consideration or a closing condition.

Q: What should happen to signing keys at closing? A: The parties should use a controlled rotation, revocation or re-enrolment process with dual control, preserved audit evidence and tested recovery. The exact method depends on architecture and mission constraints.

Q: How does software diligence affect transaction price? A: Findings affect recurring cash flow, one-time remediation, customer retention, integration cost and operational risk. The buyer should connect each material finding to a specific cash adjustment or transaction mechanic.

Q: What should the board require before approving the acquisition? A: The board should require a defined capability perimeter, traceable source and build evidence, transferable rights, operational access, supplier and people continuity, recovery evidence, customer-consent analysis and a funded first 100-day plan.

References

[1] National Institute of Standards and Technology, SP 800-218 Secure Software Development Framework Version 1.1, 2022. https://csrc.nist.gov/pubs/sp/800/218/final

[2] National Institute of Standards and Technology, SP 800-161 Revision 1 Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations, 2022. https://csrc.nist.gov/pubs/sp/800/161/r1/upd1/final

[3] National Institute of Standards and Technology, Software Security in Supply Chains Guidance, updated 2024. https://www.nist.gov/itl/executive-order-14028-improving-nations-cybersecurity/software-security-supply-chains-guidance

[4] National Institute of Standards and Technology, Cybersecurity Framework 2.0, 2024. https://www.nist.gov/cyberframework

[5] National Institute of Standards and Technology, NISTIR 8401 Satellite Ground Segment Applying the Cybersecurity Framework to Satellite Command and Control, 2022. https://csrc.nist.gov/pubs/ir/8401/final

[6] National Institute of Standards and Technology, SP 800-53 Revision 5 Security and Privacy Controls for Information Systems and Organizations. https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final

[7] National Aeronautics and Space Administration, NASA Software Engineering and Assurance Handbook NASA-HDBK-2203. https://swehb.nasa.gov/

[8] National Aeronautics and Space Administration, SWE-042 Source Code Electronic Access. https://swehb.nasa.gov/spaces/SWEHBVD/pages/102695328/SWE-042%2B-%2BSource%2BCode%2BElectronic%2BAccess

[9] National Aeronautics and Space Administration, SWE-158 Evaluate Software for Security Vulnerabilities. https://swehb.nasa.gov/spaces/SWEHBVC/pages/50888988/SWE-158%2B-%2BEvaluate%2BSoftware%2Bfor%2BSecurity%2BVulnerabilities

[10] National Aeronautics and Space Administration, SWE-206 Auto-Generation Software Inputs. https://swehb.nasa.gov/spaces/SWEHBVD/pages/102695540/SWE-206%2B-%2BAuto-Generation%2BSoftware%2BInputs

[11] National Aeronautics and Space Administration, NPR 7150.2 NASA Software Engineering Requirements. https://nodis3.gsfc.nasa.gov/displayDir.cfm?t=NPR&c=7150&s=2

[12] National Aeronautics and Space Administration, NASA-STD-8739.8 Software Assurance and Software Safety Standard. https://standards.nasa.gov/standard/NASA/NASA-STD-87398

[13] National Aeronautics and Space Administration, NASA-STD-1006A Space System Protection Standard. https://standards.nasa.gov/node/2430

[14] National Aeronautics and Space Administration, Space Security Best Practices Guide. https://www.nasa.gov/general/nasa-issues-new-space-security-best-practices-guide/

[15] Cybersecurity and Infrastructure Security Agency, Securing the Software Supply Chain Recommended Practices for Developers, 2023. https://www.cisa.gov/resources-tools/resources/securing-software-supply-chain-recommended-practices-developers

[16] Cybersecurity and Infrastructure Security Agency, Software Bill of Materials. https://www.cisa.gov/sbom

[17] Cybersecurity and Infrastructure Security Agency, Secure by Design. https://www.cisa.gov/securebydesign

[18] Cybersecurity and Infrastructure Security Agency, Federal Government Cybersecurity Incident and Vulnerability Response Playbooks. https://www.cisa.gov/news-events/news/federal-government-cybersecurity-incident-and-vulnerability-response-playbooks

[19] European Space Agency, Software Products for Mission Operations. https://esoc.esa.int/services-software

[20] European Space Agency, SPACE-SHIELD Malicious Supply Chain Capabilities and Software Vulnerabilities. https://spaceshield.esa.int/techniques/T2007/

[21] European Union Agency for Cybersecurity, ENISA Space Threat Landscape 2025. https://www.enisa.europa.eu/publications/enisa-space-threat-landscape-2025

[22] Consultative Committee for Space Data Systems, Mission Operations and Information Management Services. https://public.ccsds.org/Publications/MOIMS.aspx

[23] Consultative Committee for Space Data Systems, Security Working Group Publications. https://public.ccsds.org/Publications/Security.aspx

[24] United States Office of Space Commerce, Space Policy Directive 5 Cybersecurity Principles for Space Systems. https://space.commerce.gov/president-signs-space-cybersecurity-policy-directive/

[25] United Kingdom National Cyber Security Centre, Cyber Security Toolkit for Boards. https://www.ncsc.gov.uk/collection/board-toolkit

[26] Open Worldwide Application Security Project, Software Component Verification Standard. https://owasp.org/www-project-software-component-verification-standard/

[27] Open Worldwide Application Security Project, Software Assurance Maturity Model. https://owaspsamm.org/

[28] The Linux Foundation, SPDX Software Package Data Exchange. https://spdx.dev/

[29] International Organization for Standardization, ISO IEC 27001 Information Security Management Systems. https://www.iso.org/standard/27001

[30] European Cooperation for Space Standardization, ECSS Software Engineering Standards. https://ecss.nl/standards/active-standards/software-engineering/

About the Author

Chennakeshav (CK) is a corporate finance and investment banking executive with 25+ years of global experience in deal origination, structuring and execution across M&A, growth capital and corporate strategy. He has led value-creation mandates for founders, corporates and funds — bridging the boardroom view to hands-on execution and close.

His career spans Morgan Stanley, HSBC, Lloyds Banking Group, EWEC, ADQ portfolio companies and Emirates Growth Fund, across TMT, real estate, fintech, deeptech, cleantech, infrastructure and energy. He has partnered with C-suite leaders, private equity and venture funds, sovereign wealth funds and family offices to finance complex fund raises and scale-up ventures, and has led M&A due diligence, post-merger integration and business-transformation initiatives to create value.

At Matchpoint Partners he is Managing Partner, leading the firm's corporate finance, M&A and capital-raising practice. He holds an MBA from London Business School, an engineering degree from VTU and a Master of Laws (LLM, in progress) from UCL London.

An active start-up mentor, CK mentors at Techstars, DIFC FinTech Hive, Startup Grind, Founder Institute and IN5, serves as Entrepreneur Mentor in Residence (EMiR) at London Business School, and judges the Entrepreneurship World Cup.

https://www.linkedin.com/in/ckadya/

https://www.matchpoint-partners.com/team/ck-adya.html

This paper is part of a continuing series on the structure of private and alternative markets. The views expressed are the author's own. The paper is for information only, describes market structure in general terms, and does not constitute investment, legal, tax or regulatory advice or a recommendation in respect of any security, vehicle or counterparty.