
Community.OS - Security at Mosa

Document	Security at Mosa
Document Version	v1.0
Reviewed	27th September 2026
Status	Approved
Parent Company	Sidekick Platforms
Platform Brand	Community.OS
Consumer Product	Mosa
Brand Promise	Making Participation Visible
Core Message	When participation becomes visible, communities become stronger.
Supporting Line	Every little bit helps.
Guide	Pitch

Table of Contents

Community.OS - Security at Mosa	1
Table of Contents	2
1. Security at Mosa	4
2. Security Is More Than Data Protection	5
3. One Security Foundation Across Mosa	5
4. Identity, People and Accounts	6
5. Protecting Accounts	6
6. Your Role in Account Security	6
7. Authentication and Authority Are Different	7
8. Protecting Organisation Administrator Access	7
9. Higher-Consequence Actions Receive More Care	8
10. Sidekick Platforms Personnel Access	8
11. Support and Exceptional Access	9
12. Highly Privileged Platform Access	9
13. Protecting Information According to Consequence	10
14. Encryption	10
15. Secrets and System Credentials	11
16. Participant Conversations	11
17. Organisation Operational Communications	11
18. Communication Metadata	12
19. Reporting Harmful Communications	12
20. Children, Young People and Security	13
21. Security Logging and Auditability	13
22. Logs Should Not Become Another Sensitive Dataset	14
23. Monitoring and Detection	14
24. Building Mosa Securely	14
25. Vulnerability Management	15
26. Security Testing and Assurance	15
27. Infrastructure and Hosting	16
28. Service Providers and Supplier Security	16
29. Organisation-Controlled Integrations	17
30. APIs and Machine Access	17
31. Payment Security	17
32. Backups and Recovery	18
33. Recovery Commitments	18
34. Reliability Is Part of Trust	18
35. Safe Degradation	19
36. Security and Privacy Incidents	19
37. Known, Suspected and Confirmed Incidents	20

38. Privacy Incidents and Notification	20
39. Shared Security Responsibility	20
40. Security for Community Organisations	21
41. Reporting a Security Concern	21
42. Security and Safeguarding	21
43. Security and Privacy	22
44. Security and Artificial Intelligence	22
45. Security Across Community Spaces	23
46. Security Evolves With Mosa	23
47. Security Risk and Safe Enablement	23
48. What We Do Not Promise	24
49. Changes to Security at Mosa	24
50. Contact Us	25
Our Security Commitment	25

1. Security at Mosa

Communities rely on trust.

When you use Mosa, you may trust us with information about:

- who you are;
- the Organisations you are connected to;
- the activities you participate in;
- your qualifications;
- your communications;
- your payments;
- your family or Household relationships;
- and, in some circumstances, information that deserves additional care.

Organisations also rely on Community.OS for capabilities that matter, including:

- communicating with their communities;
- managing participation;
- maintaining records;
- connecting integrations;
- and giving authorised people appropriate access.

We take that trust seriously.

Security is part of how we design and operate Mosa.

It influences how we think about:

- identity;
- access;
- permissions;
- information;
- communications;
- infrastructure;
- recovery;
- and the people who operate Community.OS.

Security is not only about keeping information secret.

It is also about making sure the right people can do the right things, in the right context, while preventing people from gaining access or authority they should not have.

Our approach is built around a simple idea:

Protect the Person. Protect their information. Protect legitimate authority. Keep communities operating safely.

2. Security Is More Than Data Protection

Mosa is designed around people, relationships and participation.

That means security needs to protect more than stored information.

We think about security across:

- confidentiality;
- integrity;
- availability;
- identity;
- authority;
- continuity;
- participant safety;
- and community trust.

For example, a security problem may occur if:

- somebody sees information they should not see;
- somebody changes information they should not be able to change;
- somebody gains authority they should not have;
- a participant Account is compromised;
- an Organisation administrator is given excessive access;
- an integration receives more access than intended;
- or a critical part of Community.OS becomes unavailable.

3. One Security Foundation Across Mosa

Community.OS provides the underlying platform for Mosa and Organisation experiences.

We use a common security foundation across areas such as:

- participant experiences;
- Organisation experiences;
- communications;
- integrations;
- APIs;
- payments;
- files;
- infrastructure;
- and future Community Spaces.

That helps us avoid creating separate trust models for every new part of Mosa.

Different capabilities may require different controls, but they should still operate within the same underlying security and authority model.

4. Identity, People and Accounts

Mosa distinguishes between a **Person** and an **Account**.

A Person may exist within Community.OS before they have direct Account access.

For example, a child may participate through an authorised parent or representative before they control their own Account.

That means:

Person Existence ≠ Direct Account Access

and:

Direct Account Access ≠ Authority Over Everything Connected to That Person

Account access is only one part of security.

Community.OS also considers whether the Person is actually authorised to perform the requested action.

5. Protecting Accounts

Mosa uses security controls appropriate to the Account, capability and level of risk.

These may include:

- secure authentication;
- session controls;
- account recovery protections;
- verification;
- additional checks for higher-consequence actions;
- suspicious authentication detection;
- and other protections as Mosa develops.

As the platform grows, we may introduce stronger authentication methods for higher-risk capabilities, including modern approaches such as passkeys.

We do not want ordinary participants to need specialist security knowledge simply to use Mosa safely.

The safe path should increasingly be the normal path.

6. Your Role in Account Security

People using Mosa also play a role in protecting their Accounts.

You should:

- protect your authentication credentials;
- avoid sharing your Account;

-
- use your own Account rather than another Person's;
 - keep relevant recovery information current;
 - protect devices used to access Mosa;
 - and tell us if you believe your Account may have been compromised.

Where Mosa supports authorised representation, a parent, guardian or representative should use that capability rather than signing into another Person's Account as though they were that Person.

7. Authentication and Authority Are Different

Being signed in tells Community.OS something about **who is acting**.

It does not automatically establish **what they are allowed to do**.

Authority may depend on:

- Organisation;
- role;
- Group;
- relationship;
- representation;
- Community Space;
- activity;
- information category;
- safeguarding requirements;
- time;
- and other relevant context.

A Person may have significant authority in one Organisation and no authority in another.

That separation is important to Mosa.

8. Protecting Organisation Administrator Access

Organisation administrators may be trusted with capabilities that have significant consequences.

Depending on their role, administrators may be able to:

- manage memberships;
- invite participants;
- manage activities;
- communicate with communities;
- view relevant participant information;
- manage roles;
- configure Organisation capabilities;
- or perform other privileged actions.

We design administrator access so that it can be:

- individually attributable;

-
- appropriately authenticated;
 - limited to relevant capability;
 - reviewable;
 - and revocable.

Organisations are responsible for selecting appropriate administrators, periodically reviewing access and removing authority that is no longer required.

9. Higher-Consequence Actions Receive More Care

Not every action in Mosa carries the same risk.

Changing a notification preference is different from:

- exporting participant information;
- granting administrator authority;
- changing security settings;
- connecting an integration;
- accessing protected information;
- or changing another Person's authority.

Higher-consequence actions may therefore receive additional protection.

Depending on the capability, this may include:

- stronger authentication;
- confirmation;
- additional permission checks;
- audit logging;
- restricted access;
- manual approval;
- or another appropriate control.

10. Sidekick Platforms Personnel Access

Working for Sidekick Platforms does not automatically mean somebody can access participant information or production systems.

Our position is simple:

Employment ≠ Information Authority

Internal access should follow legitimate operational responsibility.

Privileged access is intended to remain:

- individually attributable;
- appropriately authenticated;
- limited to the task;
- and revocable.

As Sidekick Platforms grows, access should become increasingly aligned to role and responsibility rather than becoming broader simply because more people work on Mosa.

11. Support and Exceptional Access

There may be circumstances where authorised Sidekick Platforms personnel need elevated access to:

- diagnose a problem;
- provide authorised support;
- investigate an incident;
- recover a service;
- address a security issue;
- or respond to another legitimate operational requirement.

We distinguish between different forms of elevated access.

Diagnostic Access helps us understand a problem.

Delegated Access may be used where access is appropriately authorised for a defined support or operational purpose.

Break-Glass Access is exceptional access used where urgent circumstances justify temporarily going beyond ordinary access pathways.

Break-glass access is intended to remain exceptional, not become a convenient substitute for proper permissions.

12. Highly Privileged Platform Access

Community.OS requires some highly privileged operational capabilities.

That does not mean a highly privileged user should automatically see everything.

Our position is:

Super-Administrator ≠ Omniscient

Highly privileged capabilities should be separated according to purpose where practical.

Similarly:

Founder or Executive Status ≠ Permanent Unlimited Access

Early-stage operational responsibility may sometimes require broader access than a mature organisation eventually needs.

That access should not remain permanently broad simply because it existed during Mosa's early development.

13. Protecting Information According to Consequence

Different information and capabilities carry different levels of consequence.

Community.OS considers what could happen if information were:

- disclosed;
- changed;
- destroyed;
- made unavailable;
- or used to exercise authority improperly.

For example, a public Organisation description does not require the same treatment as:

- Health & Safety Information;
- safeguarding information;
- authentication credentials;
- payment information;
- or administrator authority.

We aim to apply protections according to the nature, context and consequence of the information or capability involved.

14. Encryption

Encryption is one of the controls used to protect information in modern systems.

Community.OS uses appropriate platform and provider capabilities to protect information in transit and at rest where relevant.

Encryption is only one part of security.

It does not replace:

- authentication;
- permissions;
- access control;
- safeguarding;
- secure development;
- monitoring;
- or responsible information handling.

A system can use strong encryption and still expose information through incorrect permissions.

Security therefore depends on the whole trust model.

15. Secrets and System Credentials

Community.OS relies on sensitive system credentials such as:

- application secrets;
- API credentials;
- service credentials;
- signing keys;
- encryption material;
- and other privileged secrets.

These are treated differently from ordinary application information.

Access should be limited to legitimate operational need.

Sensitive credentials should not be treated as ordinary configuration simply because they are used by software rather than people.

16. Participant Conversations

Mosa may support personal one-to-one and participant-created Group conversations.

These conversations have a different trust model from Organisation operational communications.

Our longer-term architecture supports stronger participant-controlled cryptographic protection for appropriate participant conversations.

However:

We do not promise that every current Mosa communication is end-to-end encrypted unless the relevant capability expressly states that it is.

We prefer to describe the protection a capability actually provides rather than make promises about future security features before they are implemented and verified.

17. Organisation Operational Communications

Organisation operational communications may require a different trust model from private participant conversations.

Examples may include:

- event notices;
- roster information;
- registration messages;
- safety communications;
- membership information;
- and other Organisation communications.

These communications may require:

- Organisation continuity;
- appropriate records;
- authorised administrator access;
- safeguarding;
- auditability;
- or operational history.

The security model for an operational communication therefore does not have to be identical to the model for a private participant conversation.

18. Communication Metadata

Protecting message content does not mean a communication system contains no other security-relevant information.

A communication service may need information such as:

- sender;
- recipient or Group;
- time;
- delivery state;
- conversation identifier;
- device information;
- abuse signals;
- or routing information.

This information can itself be sensitive.

We aim to minimise and protect communication metadata appropriately.

19. Reporting Harmful Communications

Strong communication privacy should not require Sidekick Platforms to maintain permanent access to every private conversation simply so harmful content can later be reported.

Where supported, a participant reporting harmful content may deliberately provide:

- the relevant message;
- appropriate surrounding context;
- sender information;
- or another limited evidentiary package

to an authorised reporting or moderation process.

Conceptually:

Private Conversation → Participant Reports Specific Content → Limited Disclosure for the Report

This helps preserve communication privacy while still allowing serious concerns to be raised.

20. Children, Young People and Security

Security involving children and young people requires more than protecting Account credentials.

Relevant protections may also involve:

- representation;
- communication authority;
- safeguarding;
- age and relationship context;
- restricted message initiation;
- supervised communication;
- Group communication;
- or additional Organisation safeguards.

Security should support safeguarding without assuming that every adult with an Organisation role should receive unrestricted access to a young Person.

21. Security Logging and Auditability

Community.OS maintains security and operational records appropriate to the consequence of an action.

For significant actions, records may include information about:

- who acted;
- relevant Organisation;
- what action occurred;
- affected resource;
- time;
- result;
- authentication context;
- permission or security decision;
- integration or service identity;
- and other relevant context.

This helps us understand what happened when something goes wrong.

We place particular importance on visibility around:

- privilege changes;
- unusual authentication activity;
- integrations;
- bulk access;
- and significant production changes.

22. Logs Should Not Become Another Sensitive Dataset

Logging everything is not automatically good security.

Security logs should avoid unnecessarily recording information such as:

- passwords;
- authentication secrets;
- session tokens;
- encryption keys;
- message content;
- sensitive health information;
- safeguarding content;
- payment credentials;
- or identity evidence.

Where possible, security telemetry should tell us **what happened** without unnecessarily recreating the sensitive information being protected.

23. Monitoring and Detection

Community.OS may monitor systems and security events to identify potential:

- Account compromise;
- suspicious authentication;
- unauthorised access;
- unusual privilege changes;
- abnormal bulk activity;
- integration abuse;
- malicious behaviour;
- service disruption;
- or other security concerns.

A security signal does not automatically mean malicious conduct occurred.

Context matters.

For example, accessing many participant records may be ordinary for an authorised administrator of a large Organisation and unusual for somebody performing a much smaller role.

As Mosa matures, our security detection should increasingly take this context into account.

24. Building Mosa Securely

Security is considered throughout the development and operation of Community.OS.

Our practices continue to mature with the platform and may include:

- code review;

- dependency management;
- secure configuration;
- testing;
- permission testing;
- environment separation;
- controlled deployment;
- security review of consequential features;
- and regression protection for important security fixes.

Security is not limited to traditional software vulnerabilities.

A business-logic failure can also be a security issue.

For example, a defect that accidentally allows one Organisation to see another Organisation's information is a security failure even if no traditional software exploit was involved.

25. Vulnerability Management

No meaningful software platform should assume vulnerabilities will never be discovered.

When a potential weakness is identified, our approach is based around:

Discover → Record → Triage → Contain Where Necessary → Prioritise → Remediate → Verify → Close → Learn

A vulnerability is not necessarily resolved merely because a code change has been deployed.

Important fixes should be verified and, where practical, protected against regression.

26. Security Testing and Assurance

As Mosa develops, our security assurance will continue to mature according to platform scale, information and risk.

This may include:

- automated testing;
- dependency and vulnerability scanning;
- access-control testing;
- configuration review;
- security review;
- penetration testing;
- provider assurance;
- incident exercises;
- recovery testing;
- and independent assurance where appropriate.

We do not claim a certification, penetration-testing regime or assurance level unless that assurance actually exists.

Security claims should be supported by real capability and evidence.

27. Infrastructure and Hosting

Community.OS relies on specialist technology providers for parts of its infrastructure and service delivery.

Using specialist providers can improve:

- security;
- resilience;
- scalability;
- and operational maturity.

It does not remove our responsibility for how Community.OS uses and configures those services.

Our approach is:

Outsourced Service ≠ Outsourced Accountability

and:

Provider Security ≠ Community.OS Configuration of Provider

A secure infrastructure provider can still be used insecurely.

We remain responsible for the Community.OS systems and configurations within our control.

28. Service Providers and Supplier Security

Service providers may support areas such as:

- hosting;
- databases;
- communications;
- payments;
- monitoring;
- development infrastructure;
- support;
- and other platform capabilities.

We assess consequential providers according to the trust placed in them.

Relevant considerations may include:

- security;
- privacy;
- reliability;
- information handled;
- data location;
- subprocessors;
- configuration;

-
- criticality;
 - concentration risk;
 - and ability to replace a provider where necessary.

Our provider assurance should continue as material circumstances change rather than being treated as a one-time selection exercise.

29. Organisation-Controlled Integrations

An Organisation may connect Community.OS to systems or providers it selects or controls.

An Organisation-controlled provider is not necessarily a Sidekick Platforms provider.

Responsibility therefore depends on who controls the relevant system.

Sidekick Platforms remains responsible for Community.OS capabilities within our control.

The Organisation remains responsible for matters within its control, including:

- selecting appropriate external systems;
- managing Organisation-controlled credentials;
- securing Organisation-controlled devices;
- configuring integrations appropriately;
- managing access within external systems;
- and protecting information after it has legitimately moved into an Organisation-controlled environment.

Security responsibility follows meaningful control.

30. APIs and Machine Access

Integrations and automated systems may require access without a human being actively signed in.

Machine access should still have:

- an identifiable purpose;
- appropriate credentials;
- defined permissions;
- appropriate scope;
- revocation capability;
- and security monitoring.

An API credential should not become a technical super-administrator simply because it is used by software.

Read and write authority should be separated where appropriate.

31. Payment Security

Where Mosa supports payments, we prefer to use appropriate specialist payment infrastructure rather than unnecessarily holding sensitive payment credentials ourselves.

Different parts of a payment may be controlled by:

- Mosa;
- an Organisation;
- a payment provider;
- or another financial-service provider.

Responsibility should reflect the system and transaction component each party controls.

32. Backups and Recovery

Security includes being able to recover from failure.

Community.OS uses backup and recovery capabilities appropriate to the systems and information being protected.

Backups themselves require protection.

Recovery is not simply about restoring data somewhere.

It can also involve checking whether:

- information is complete;
- permissions remain correct;
- integrations are consistent;
- transactions are reconciled;
- and the recovered system can safely return to operation.

33. Recovery Commitments

Different systems may require different recovery objectives depending on their consequence.

We do not publish arbitrary recovery-time or recovery-point promises simply because those numbers sound reassuring.

Where specific recovery or service commitments are made, they should reflect:

- actual architecture;
- tested capability;
- operational evidence;
- customer requirements;
- and applicable contractual commitments.

Enterprise arrangements may include specific service commitments where appropriate.

34. Reliability Is Part of Trust

Security and reliability are closely connected.

A platform that protects information but cannot reliably support participation is not delivering the whole trust requirement.

Our approach to service resilience considers the lifecycle:

Prevent → Detect → Contain → Degrade Safely → Recover → Reconcile → Verify → Communicate → Learn

The objective is not to claim that Mosa will never experience a failure.

It is to be able to detect, contain, recover from and learn from failures.

35. Safe Degradation

When something goes wrong, the safest response is not always to keep every feature operating.

In some circumstances, Community.OS may temporarily:

- disable a capability;
- prevent a consequential action;
- limit an integration;
- restrict an Account;
- delay processing;
- operate in a reduced-function state;
- or make part of Mosa temporarily unavailable.

A temporary reduction in functionality may be safer than allowing uncertain or unsafe behaviour to continue.

36. Security and Privacy Incidents

If we become aware of a suspected privacy or security incident, we assess and respond according to its nature and consequence.

Our incident approach includes:

Detect → Triage → Contain → Stabilise → Recover → Reconcile → Communicate → Learn → Close

Depending on the incident, our response may involve:

- investigation;
- containment;
- protecting affected systems or information;
- restricting compromised access;
- credential changes;
- recovery;
- reconciliation;
- correcting information where necessary;
- preserving relevant evidence;
- determining legal or regulatory obligations;
- communicating with affected people or Organisations where appropriate;

- and reducing the likelihood or consequence of recurrence.

37. Known, Suspected and Confirmed Incidents

Security investigations evolve.

We distinguish between information that is:

Known

Suspected

Under Investigation

and:

Confirmed

We aim to communicate according to verified facts rather than prematurely presenting assumptions as conclusions.

38. Privacy Incidents and Notification

A security event is not automatically a notifiable privacy breach.

Where an incident involves personal information, we assess applicable privacy obligations.

Where the Australian Notifiable Data Breaches scheme or another applicable notification requirement applies, we will comply with those obligations.

Our Privacy Policy provides further information about our approach to privacy and security incidents.

39. Shared Security Responsibility

Security responsibility follows meaningful control.

Sidekick Platforms is responsible for the security of Community.OS capabilities within our control.

Organisations are responsible for matters within their control.

This may include:

- selecting appropriate administrators;
- protecting Organisation-controlled credentials;
- periodically reviewing administrator access;
- removing access that is no longer required;
- securing Organisation-controlled devices and systems;
- using participant information appropriately;
- managing Organisation-controlled integrations;

- and responding to Organisation-specific security concerns.

The fact that Mosa connects these systems does not mean one party automatically inherits every security responsibility of the other.

40. Security for Community Organisations

Many Organisations using Mosa will not have:

- a Chief Information Security Officer;
- a security team;
- specialist IT staff;
- or enterprise security infrastructure.

They should not need those things simply to operate a local club, school community, charity or community group safely.

Community.OS should make good security behaviour easier through platform design.

Security should support community participation rather than turning every volunteer administrator into a cybersecurity specialist.

41. Reporting a Security Concern

If you believe:

- your Account has been compromised;
- somebody has accessed information without authority;
- an Organisation administrator Account has been compromised;
- an integration credential may have been exposed;
- you have discovered a potential security vulnerability;
- or another serious security concern affects Mosa,

please tell us promptly.

Until a dedicated security reporting channel or responsible disclosure process is published, security concerns may be sent to:

hello@oncommunityos.com

Please do not publicly disclose sensitive security information in a way that unnecessarily exposes participants or Community.OS to additional risk.

42. Security and Safeguarding

Security and safeguarding overlap, but they are not identical.

A system can be technically secure while still allowing inappropriate authority.

For example:

**Correctly Authenticated Coach + Inappropriate Private Communication Authority =
Safeguarding Failure**

Likewise, a safeguarding control can also become an important security boundary.

Community.OS therefore treats:

- identity;
- authority;
- privacy;
- security;
- and safeguarding

as connected parts of the trust model.

43. Security and Privacy

Security and privacy are related, but different.

Privacy asks:

Should this actor receive this information for this purpose?

Security asks:

How do we make sure only the properly authorised actor can perform the permitted action?

Strong security does not justify unnecessary access.

Likewise, good privacy intentions are not enough if technical controls do not enforce them.

44. Security and Artificial Intelligence

Where Community.OS uses artificial intelligence or AI-assisted services, those capabilities remain subject to the same security and trust principles as other systems.

The security consequences depend on how AI is used.

Development assistance is different from:

- internal operational assistance;
- participant-facing AI;
- AI processing participant information;
- or AI model training.

Providers receiving consequential information should be subject to appropriate trust assessment.

45. Security Across Community Spaces

Community.OS may expand across many Community Spaces.

New Spaces may introduce:

- new information;
- new transactions;
- new integrations;
- new authority;
- and new security risks.

They should still operate within the same underlying security foundation.

A new Community Space may require additional controls, but it should not quietly create a separate trust model.

46. Security Evolves With Mosa

The security appropriate for an early platform will not necessarily be sufficient when Mosa supports:

- significantly more people;
- more Organisations;
- more sensitive information;
- additional jurisdictions;
- more integrations;
- payments;
- messaging;
- AI capabilities;
- or more consequential Community Spaces.

Our security capability should continue to mature as Mosa grows and as the consequences of failure change.

47. Security Risk and Safe Enablement

No useful technology platform can eliminate every possible risk.

The objective is not maximum restriction.

It is **safe enablement**.

We aim to understand meaningful risks, reduce them proportionately and make deliberate decisions about significant residual risk.

Security should help Mosa deliver useful community capability safely rather than becoming an automatic reason to prevent useful functionality.

48. What We Do Not Promise

No internet-connected service can responsibly guarantee absolute security, uninterrupted availability or that a security incident will never occur.

We do not describe Mosa as:

- unhackable;
- perfectly secure;
- permanently available;
- immune from human error;
- or free from every possible vulnerability.

Instead, our objective is to build and operate security capabilities appropriate to:

- the information we handle;
- the authority Community.OS enables;
- the people and communities using Mosa;
- the consequence of failure;
- and the scale and maturity of the platform.

Where we make specific security or service commitments, those commitments should be supported by actual capability and evidence.

49. Changes to Security at Mosa

Security practices will evolve as Community.OS develops.

We may update this document to reflect:

- new capabilities;
- new security controls;
- material platform changes;
- new assurance;
- changes to providers;
- regulatory developments;
- or improvements arising from incidents and security learning.

The current version and last-updated date will be identified.

Some more detailed security information may also be made available through the Mosa Trust Centre as that capability develops.

50. Contact Us

Mosa and Community.OS are operated by:

OSA Management Pty Ltd
ABN 62 677 945 863
Trading as Sidekick Platforms

For security questions or concerns:

hello@oncommunityos.com

A dedicated security contact and responsible disclosure process may be introduced as Mosa develops.

Our Security Commitment

Communities should be able to use Mosa with confidence.

That means protecting more than information.

It means protecting people, legitimate authority, access, participation and the ability for communities to keep operating when something goes wrong.

We believe:

Identity should be protected.

Authority should come from legitimate context, not simply from a title.

Access should follow responsibility.

Employment should not mean unrestricted visibility.

Administrators should have the capabilities they genuinely need, not everything the platform can technically expose.

Sensitive information should not appear in places simply because collecting or logging it is convenient.

Organisations should be able to operate safely without becoming cybersecurity experts.

Specialist providers can help us build a stronger platform, but using them does not remove our responsibility for how Community.OS is configured and operated.

Security failures should be detected, contained, recovered from and learned from.

Security claims should follow real capability and evidence.

Most importantly:

Security should protect participation, not prevent it.

Our aim is not to promise that nothing will ever go wrong.

It is to operate Mosa responsibly, strengthen security as the platform grows, treat consequential authority carefully, respond appropriately when something does go wrong and continue earning the trust communities place in us.