International Journal of Technology Management & Humanities (IJTMH)
e-ISSN: 2454 – 566X, Volume 6, Issue 3-4, (July-December 2020), www.ijtmh.com

Strengthening Hybrid SAP Ecosystems Through Identity Federation, Intelligent Access Management, and Zero-Trust API Security

Pradeep Tiwari
Independent Researcher
Tiwari.pradeep04@gmail.com

Abstract

Hybrid SAP ecosystems are only as secure as the identities, roles and interfaces that connect them, and each of those three layers is now stretched across on-premise SAP ECC, cloud applications, mobile Fiori front ends and automated bots. This article argues that the three layers must be redesigned together: identity federation to give every human and machine actor one trusted identity, intelligent access management to keep authorizations lean and free of toxic combinations, and zero-trust API security to verify every call between systems instead of trusting the network.

The argument draws on seven studies published up to August 2020. Three by Asuri (2019a, 2019b, 2020) examine SAP authorization redesign, Fiori and embedded analytics expansion in SAP ECC, and RPA- and machine-learning-driven workflow automation. Four foundational works supply the security model: Kindervag's (2010) zero-trust model, Google's BeyondCorp deployment (Ward & Beyer, 2014), NIST's attribute-based access control guidance (Hu et al., 2014), and NIST's Zero Trust Architecture (Rose et al., 2020). From these, the article derives a layered reference framework and a phased roadmap for SAP security teams.

Keywords: SAP ECC, hybrid ERP, identity federation, single sign-on, segregation of duties, role-based and attribute-based access control, robotic process automation, SAP Fiori, OData, API security, zero trust.

1. Introduction

The classic SAP security model assumed a closed system: users logged on through SAP GUI from inside the corporate network, and authorization roles were the main line of defence. That model no longer matches how SAP is run. A typical landscape today combines an SAP ECC or S/4HANA core, cloud applications for HR, procurement or customer engagement, a Fiori launchpad reached from browsers and phones, middleware that moves data between systems, and software robots that post documents on behalf of people.

Each of these extensions widens the attack surface in a different way. Fiori and embedded analytics move SAP transactions into a web channel built on OData services, which turns business functions into callable APIs (Asuri, 2019b). Robotic process automation and machine learning add non-human actors that log on, read data and post transactions at machine speed (Asuri, 2020). Migration and consolidation projects inherit years of accumulated roles, many of which grant risky combinations of access that nobody intended (Asuri, 2019a).

The perimeter, in other words, has dissolved. Kindervag (2010) argued that a trusted internal network is itself the weakness, since an attacker or insider who gets inside meets little further resistance. Ward and Beyer (2014) showed that a large enterprise could move access decisions away from the network and onto verified users and devices. Rose et al. (2020) formalised the idea in NIST's Zero Trust Architecture: no

request is trusted because of where it comes from, and every access is decided per session against identity, device and context.

This article applies that thinking to hybrid SAP ecosystems. It asks three questions:

1. How can identity federation give every user, service and bot a single, verifiable identity across on-premise and cloud SAP systems?
2. How can access management become intelligent, meaning risk-aware, consolidated and continuously monitored, rather than a static list of roles?
3. How can the APIs that now carry SAP business processes be secured under zero-trust principles?

Section 2 reviews the seven studies. Sections 3 to 5 take each question in turn. Section 6 combines them into a reference framework and roadmap, and Section 7 sets out limitations and open problems.

2. Literature review

The seven studies fall into two groups. Asuri's three papers describe how SAP landscapes are changing from the inside: roles are being redesigned, user interfaces are moving to the web, and processes are being automated. The four security works describe the model that those changes now require.

2.1 SAP authorization redesign

Asuri (2019a) treats the SAP authorization concept as an architecture problem rather than a maintenance task. The study links role design to business process risk assessment, so that the processes with the highest financial or compliance exposure shape how access is structured. It gives particular attention to toxic access combinations, the segregation-of-duties conflicts in which one user can, for example, both create a vendor and pay it. It also frames role consolidation as part of migration planning, arguing that the move to a new platform is the moment to reduce role sprawl rather than copy it forward.

For a hybrid ecosystem, this work supplies the inner layer of control. Federation and zero trust decide who is allowed in; the authorization architecture decides what they can do once inside.

2.2 Fiori expansion and embedded analytics

Asuri (2019b) examines how SAP Fiori apps and embedded analytics can modernise legacy SAP ECC, improving user experience and operational performance without a full platform replacement. Fiori delivers role-based, task-focused apps through a browser-based launchpad, and embedded analytics places operational reporting close to the transaction.

The security consequence is structural. Fiori apps communicate with the back end through OData services exposed by the SAP Gateway. Each new app therefore adds web-facing endpoints, front-end catalogues and back-end service authorizations that must be kept in step. A user experience programme is, in effect, also an API programme.

2.3 RPA and machine learning in SAP workflows

Asuri (2020) proposes a framework that combines robotic process automation with machine learning to automate standard SAP ECC business processes. Bots handle rule-based steps, while machine-learning components handle classification and decision tasks that once needed human judgement.

This creates a new population of identities. Bots need SAP logons, often with broad access spanning several process steps, and they act without the natural checks of a human user. Unless bot identities are designed with the same rigour as human ones, automation can quietly recreate the toxic combinations that role redesign set out to remove.

2.4 Zero trust and modern access control

Kindervag (2010) introduced the zero-trust model at Forrester Research. Its core claim is that networks should have no trusted interior: all traffic should be inspected and logged, and access should be granted on a strict least-privilege basis regardless of source.

Ward and Beyer (2014) described Google's BeyondCorp, which removed the privileged corporate network as the basis for access. Access to internal applications depended instead on what was known about the user and the device, enforced by an access proxy in front of each application. BeyondCorp showed that zero trust could be operated at scale, not only argued for.

Hu et al. (2014), in NIST Special Publication 800-162, defined attribute-based access control (ABAC). Under ABAC, an access decision evaluates attributes of the subject, the object, the requested operation and the environment against policy. This makes it possible to express rules that roles alone cannot, such as limiting a high-value payment approval to a managed device during working hours.

Rose et al. (2020), in NIST Special Publication 800-207, set out a reference zero-trust architecture. Its central components are a policy engine that decides on each access request, a policy administrator that carries out the decision, and policy enforcement points that sit in front of resources. It treats every enterprise asset, including services and APIs, as a resource to be protected individually.

2.5 Gap in the literature

The SAP-focused studies describe change inside the ERP landscape but do not set out a zero-trust model for it. The security studies describe zero trust in general terms but not for an ERP with its own role model, gateway services and business-process risks. This article joins the two.

3. Identity federation across on-premise and cloud SAP

Identity federation is the foundation of the model: every later control depends on knowing exactly who, or what, is making a request. In a hybrid SAP landscape, the goal is one authoritative identity per actor, issued by a central identity provider and trusted by every SAP and non-SAP system through standard protocols.

3.1 Why fragmented identity is the first risk

Many SAP estates still hold separate user masters in each system and client, with passwords managed locally. The same person may exist under different user IDs in ECC, a cloud HR system and a procurement platform. Leavers are removed from some systems and missed in others. Under these conditions, neither access reviews nor zero-trust decisions can be reliable, because the identity they rest on is ambiguous.

3.2 The federated model

A federated design separates authentication from the SAP systems themselves:

- **A central identity provider** authenticates users once, applies multi-factor authentication and device checks, and issues signed assertions or tokens.
- **SAP systems act as service providers.** The Fiori launchpad and web applications accept SAML 2.0 assertions; SAP GUI can use Kerberos-based single sign-on inside the domain; cloud applications accept SAML or OpenID Connect.
- **Identity lifecycle is automated.** Joiner, mover and leaver events from the HR system drive provisioning and de-provisioning across every connected system, so that a change in one place is reflected everywhere.
- **Principal propagation** carries the end user's identity from the front end through middleware to the SAP back end, so that a back-end call runs under the real user rather than a shared technical account.

This mirrors the BeyondCorp approach, in which an application trusts the identity and device information presented to it rather than the network the request arrived on (Ward & Beyer, 2014).

3.3 Federating the expanded front end

The Fiori expansion described by Asuri (2019b) makes federation more urgent. Users reach Fiori from browsers outside the corporate network and from mobile devices, which is exactly the setting in which password-based SAP logon is weakest. Placing the launchpad behind the corporate identity provider brings Fiori access under the same multi-factor and conditional-access policies as every other business application. Embedded analytics benefits in the same way: a user sees operational figures only after a strong, central authentication.

3.4 Machine and bot identities

The automation framework of Asuri (2020) introduces actors that cannot use a browser login or answer a multi-factor prompt. They still need federated, governed identities. Practical principles include:

- Each bot gets its own named identity, never a shared or reused dialog user.
- Each bot identity has a documented human owner who is accountable for its access.
- Bots authenticate with certificates or short-lived tokens issued through the identity platform, not long-lived passwords stored in scripts.
- Bot identities are included in the same lifecycle and recertification processes as human users.

Treating bots as first-class identities is what makes the later access and API controls apply to them at all.

4. Intelligent access management

Intelligent access management means authorizations that are designed from business risk, kept free of toxic combinations, refined by context at the moment of use, and monitored continuously. It turns the SAP role model from a static inventory into a living control.

4.1 Risk-led role architecture

Asuri (2019a) provides the starting point: roles should be derived from a business process risk assessment. In practice this means mapping each critical process, such as procure-to-pay, order-to-cash or record-to-report, to its sensitive steps, then defining the access that each job function genuinely needs. A role built this way can be explained in business terms, which makes it far easier for process owners to approve and recertify.

4.2 Eliminating toxic combinations

A toxic combination is a set of permissions that is harmless alone but dangerous together. In SAP these are usually segregation-of-duties conflicts expressed at the level of transactions, Fiori apps and authorization objects. An intelligent approach:

- Maintains a ruleset of conflicting functions, tied to the business risks they create.
- Checks every role design, role change and access request against that ruleset before it is granted.
- Requires documented mitigating controls, such as a monitored report or a second approval, where a conflict must be accepted.
- Extends the same checks to Fiori catalogues and OData service authorizations, not only classic transaction codes.
- Applies the same rules to bot identities, because an automated process that both creates and approves a document is as risky as a person who does so.

The last two points join Asuri (2019a) to the later studies. Fiori expansion (Asuri, 2019b) and RPA (Asuri, 2020) each add new routes to the same sensitive functions, and a ruleset that sees only one route leaves the others open.

4.3 Consolidation during migration

Role consolidation is most effective when it is tied to a migration or modernisation project (Asuri, 2019a). Older SAP systems often carry thousands of roles, many unused or near-duplicates. Before migration, usage data can show which roles and permissions are actually exercised; unused access is removed, overlapping roles are merged, and the clean set is what moves to the new platform. This reduces the attack surface and the effort of every later review.

4.4 Adding context with attributes

Roles answer the question "what may this job do?" They cannot answer "should this request be allowed now, from here, on this device?" ABAC fills that gap by evaluating attributes of the user, the resource, the action and the environment (Hu et al., 2014). A hybrid approach keeps roles as the stable baseline and adds attribute-based conditions for the riskiest actions. Examples include:

- Allowing approval of payments above a threshold only from a managed device.
- Restricting access to a company code to users whose HR record assigns them to it.
- Requiring step-up authentication when a sensitive Fiori app is opened from outside the network.

The identity provider from Section 3 supplies many of these attributes, which is one reason federation must come first.

4.5 Machine learning for continuous assurance

Asuri (2020) applies machine learning to automate SAP process steps. The same family of techniques can support access governance. Usage logs can be analysed to suggest role clean-up, to flag users whose actual activity departs from their peers, and to prioritise high-risk items in access reviews. Such models should support human reviewers rather than replace them, and their recommendations need to be explainable to auditors.

The result is a cycle: risk defines the roles, rules keep them clean, attributes refine each decision, and analytics show where the design is drifting.

5. Zero-trust API security for SAP integrations

In a hybrid landscape, most business transactions now travel over APIs: OData calls from Fiori, web services between SAP and cloud applications, and remote function calls from middleware and bots. Zero-trust API security means that each of these calls is authenticated, authorised and inspected on its own merits, with no trust granted because the caller sits on an internal network.

5.1 Why SAP interfaces need zero trust

SAP integrations have often relied on network location and shared technical users. A middleware system calls the back end with one powerful service account; a firewall rule allows traffic from a trusted subnet. Kindervag (2010) identified this pattern as the core weakness of perimeter security: once inside, an attacker inherits the trust given to the network. In SAP terms, compromise of one integration account can expose every function that account can reach.

The Fiori expansion in Asuri (2019b) and the automation in Asuri (2020) both increase API traffic sharply. Each Fiori app adds OData services; each bot adds calls into SAP. The interface layer has become a main route into the system and must be protected as such.

5.2 Applying the NIST zero-trust components

Rose et al. (2020) describe a policy engine, a policy administrator and policy enforcement points. Mapped to SAP:

- **Policy enforcement points** sit in front of SAP interfaces: an API gateway or management layer in front of Gateway OData services and integration endpoints, and a reverse proxy or web dispatcher in front of the Fiori launchpad.
- **The policy engine** decides each request using the caller's identity from the federation layer (Section 3), the authorization and attribute rules from the access layer (Section 4), and signals such as device state, location and recent behaviour.
- **The policy administrator** issues or refuses short-lived session credentials accordingly, and can revoke them when risk changes.

5.3 Core API controls

- **Token-based, per-request authentication.** Callers present short-lived tokens issued by the identity platform, such as OAuth 2.0 access tokens or SAML-based tokens, rather than stored passwords. Tokens are scoped to the specific services and operations needed.
- **User context end to end.** Principal propagation (Section 3.2) lets the SAP back end apply the real user's authorizations, so a shared integration account cannot be used to reach more than the user could.
- **Least-privilege service authorizations.** Each OData service and remote-enabled function is activated only where needed and protected by back-end authorization checks, in line with the role design of Section 4.
- **Mutual TLS between systems.** Both client and server prove their identity with certificates, so that system-to-system trust rests on verified identities, not on network addresses.
- **Input validation, rate limiting and threat protection** at the gateway, to block injection, abuse of mass-data endpoints and automated scraping.

- **Micro-segmentation.** Network paths are opened only between the specific systems that must talk, reducing lateral movement if one component is compromised.

5.4 Bots as API clients

Under this model, an RPA bot is simply another API client. It authenticates with its own credential, receives a scoped token, and is checked against the same segregation-of-duties rules as a person. Its calls are logged against its named identity. This answers the risk raised in Section 2.3: automation adds speed without adding unchecked privilege.

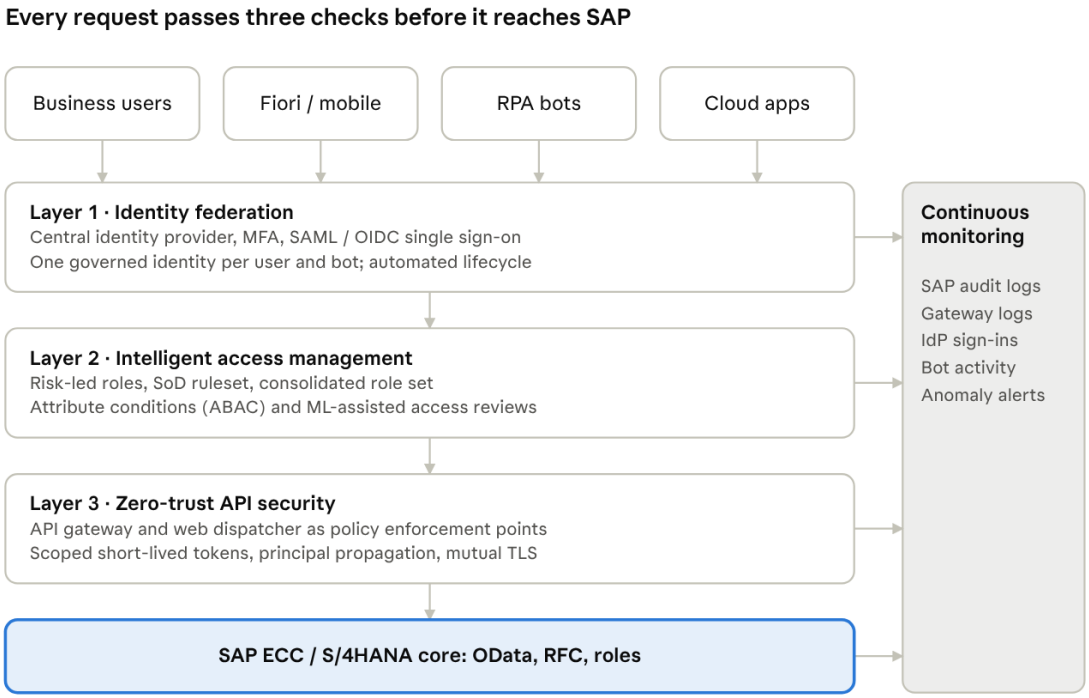
5.5 Continuous monitoring

Zero trust assumes that a breach is possible and requires constant verification (Rose et al., 2020). For SAP, this means collecting security audit logs, gateway logs, API-management logs and identity-provider sign-in records into one monitoring platform. Correlating them shows patterns that no single log reveals, such as a bot calling services outside its normal process, or a user's token used from two locations at once. The embedded analytics capability that Asuri (2019b) describes for operations can, in principle, be paired with this security data to give process owners a view of risk alongside performance.

6. Integrated reference framework and implementation roadmap

The three capabilities work only as a stack: federation establishes who is asking, access management decides what they may do, and API security enforces that decision on every call. A monitoring plane runs alongside all three and feeds what it learns back into policy.

6.1 The layered framework



Layered zero-trust framework for hybrid SAP · 3 layers, 1 monitoring plane

Requests from people, Fiori clients, bots and cloud applications pass identity, authorization and API checks in turn before touching the SAP core, while every layer reports to continuous monitoring.

Each layer maps to the studies reviewed. Layer 1 applies the BeyondCorp and NIST principle that identity, not network location, is the basis for trust (Ward & Beyer, 2014; Rose et al., 2020), extended to the Fiori users and bots introduced by Asuri (2019b, 2020). Layer 2 builds on the risk-led authorization architecture of Asuri (2019a), refined with ABAC (Hu et al., 2014). Layer 3 places NIST's policy enforcement points in front of SAP's interfaces, removing the implicit internal trust that Kindervag (2010) warned against.

6.2 Phased roadmap

1. **Assess.** Inventory all SAP users, bots, technical accounts and interfaces. Run a business process risk assessment and identify current toxic combinations (Asuri, 2019a).
2. **Federate.** Connect SAP front ends and cloud applications to a central identity provider with multi-factor authentication. Automate joiner, mover and leaver processes and give every bot a named, owned identity.
3. **Redesign and consolidate.** Rebuild roles around risk, remove unused access, enforce the segregation-of-duties ruleset across transactions, Fiori catalogues, OData services and bots, and align the work with any migration programme.
4. **Enforce at the interface.** Place an API gateway in front of OData and integration endpoints, replace shared integration passwords with scoped tokens and certificates, enable principal propagation and segment the network.
5. **Add context.** Introduce attribute-based conditions and step-up authentication for the highest-risk transactions.
6. **Monitor and learn.** Centralise SAP, gateway and identity logs, apply analytics to usage and behaviour, and feed findings back into role design and policy.

The order matters. Contextual policies and analytics depend on clean identities and roles, so attempting them first produces noisy results and little real protection.

7. Challenges, limitations and future directions

The framework is demanding to put in place, and several obstacles recur.

- **Legacy protocols.** Older SAP interfaces and custom programs may rely on fixed technical users or protocols that cannot carry tokens or user context. These need wrapping behind a gateway, re-engineering, or documented compensating controls.
- **Performance and availability.** Every additional check sits on the path of business transactions. Policy decisions must be fast and the identity provider highly available, or security will be blamed for slower month-end closes.
- **Rule quality.** Segregation-of-duties rulesets and attribute policies are only as good as the business analysis behind them. A poor ruleset produces either false alarms that reviewers ignore or gaps that auditors find.
- **Governance of machine learning.** Models that recommend access changes or flag anomalies must be explainable, monitored for drift and subject to human approval, particularly where audit evidence is required.

- **Skills and ownership.** The framework spans SAP Basis, security, identity, network and business teams. Without clear ownership, each assumes another layer is protecting the system.

The sources also impose limits on this article. Asuri's studies focus on SAP ECC; the principles carry forward to S/4HANA and SAP cloud services, but product-level details differ. The security works are vendor-neutral and predate several later developments in SAP's own identity and integration services. The framework is conceptual and has not been tested empirically here.

Useful directions for further research include:

1. Measuring how far risk-led role consolidation (Asuri, 2019a) reduces segregation-of-duties conflicts after migration.
2. Designing attribute-based policies that combine SAP authorization objects with identity-provider attributes, and testing their effect on transaction performance.
3. Defining a reference governance model for RPA and machine-learning identities in ERP systems, building on Asuri (2020).
4. Evaluating behaviour analytics on combined SAP, gateway and identity logs as an early-warning signal for fraud and account compromise.

8. Conclusion

Hybrid SAP ecosystems cannot be secured by roles alone or by a network perimeter alone. The changes described by Asuri, namely risk-led authorization redesign (2019a), Fiori and embedded analytics on SAP ECC (2019b), and RPA with machine learning (2020), each bring business value, and each adds identities, endpoints or privileges that the old model did not anticipate.

The zero-trust literature, from Kindervag (2010) through BeyondCorp (Ward & Beyer, 2014), ABAC (Hu et al., 2014) and NIST's Zero Trust Architecture (Rose et al., 2020), supplies the principle that resolves this: verify every actor and every request, grant the least access needed, and watch continuously.

Applied to SAP, that principle becomes three linked layers. Identity federation gives every person and bot one governed identity. Intelligent access management keeps what those identities can do lean, conflict-free and sensitive to context. Zero-trust API security makes sure every call between systems is proven and limited. Built together and in that order, they let organisations modernise their SAP landscape without widening its risk.

References

Asuri, R. (2019a). Redefining SAP authorization architectures through business process risk assessment, toxic access combination analysis, and migration-oriented role consolidation. *International Journal of Humanities and Information Technology*, Special 3, 186–210.

Asuri, R. (2019b). Transforming legacy enterprise resource planning systems through SAP Fiori expansion and embedded analytics to enhance user experience and operational performance in SAP ECC. *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, 11(02), 176–190. https://doi.org/10.18090/samriddhi.v11i02.13

Asuri, R. (2020). An intelligent enterprise workflow automation framework for SAP ECC standard business processes through the integration of robotic process automation and machine learning

techniques. *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, 12(02), 181–195. https://doi.org/10.18090/samriddhi.v12i02.17

Hu, V. C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R., & Scarfone, K. (2014). *Guide to attribute based access control (ABAC) definition and considerations* (NIST Special Publication 800-162). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-162

Kindervag, J. (2010). *No more chewy centers: Introducing the zero trust model of information security*. Forrester Research.

Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020, August). *Zero trust architecture* (NIST Special Publication 800-207). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207

Ward, R., & Beyer, B. (2014). BeyondCorp: A new approach to enterprise security. *;login: The USENIX Magazine*, 39(6), 6–11.