Privitty

Secure Edge Access & Industrial Data Integration

Whitepaper -- Architecture, Security, and Deployment

Version: Draft 0.1
Prepared by: Alanring Technologies -- “Privitty”
https://privitty.com

“Machine data flows freely -- access never does”

Table of Contents

Executive Summary

Industrial operators increasingly need to expose human–machine interfaces (HMIs) and programmable controllers to remote engineers and monitoring systems -- without opening the plant network to the public internet. Privitty is an edge-native, end-to-end encrypted platform purpose-built for this problem: secure remote access to factory-floor systems and controlled, auditable transfer of industrial data and program files.

This whitepaper describes two complementary deployment patterns validated against real industrial automation environments:

- Secure remote HMI visualization and data integration -- Privitty Edge runs on a gateway device co-located with an industrial HMI operating in transparent gateway mode, relaying both screen access (VNC) and structured tag data (OPC UA) to remote operators.
- Secure operator-to-factory program transfer -- Privitty Edge runs on an industrial edge PC at the factory, giving remote engineers encrypted file transfer and full remote-desktop access (RDP/VNC) for deploying, verifying, and commissioning automation projects.

In both patterns, no inbound firewall ports are opened on the factory network. All connectivity from the edge is outbound-only, and operators reach factory systems exclusively through Privitty's end-to-end encrypted (E2EE) tunnels and relay infrastructure.

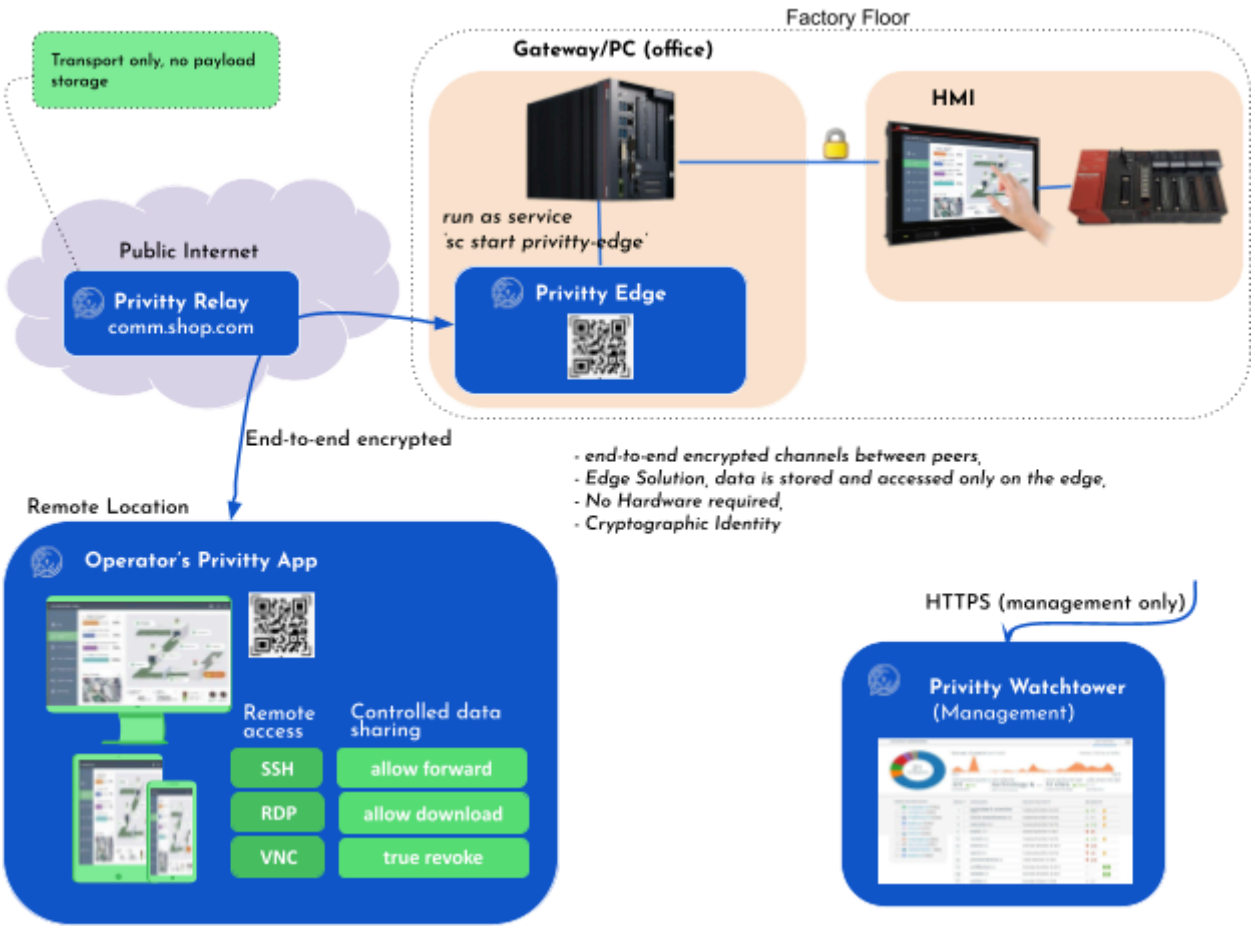
Key capabilities at a glance

Capability	How Privitty delivers it
Secure remote HMI visualization	Operator reaches the HMI's native VNC server through an E2EE tunnel, without exposing VNC directly to the internet
Secure industrial data path	An OPC UA client on the edge gateway reads tags from the HMI's built-in OPC UA server and relays them to remote systems via REST API
Secure program transfer	Encrypted file transport with per-file access control, forward restriction, expiry, and true revoke
Remote engineering access	E2EE RDP/VNC tunnels into edge-hosted engineering tools and remote-HMI client software
No cloud data retention	Production programs and session payloads remain on the edge; the relay is transport-only
Enterprise governance	Dedicated Privitty Watchtower for device, operator, and policy administration
No additional hardware	Software deployment on existing industrial PCs / Windows IoT Enterprise gateways

Physical & Network Topology

Network assumptions

- The Privitty Edge gateway and the industrial HMI share the same Layer-2/Layer-3 shop-floor subnet (direct Ethernet connectivity).
- The HMI operates in transparent gateway mode, allowing the gateway and other Ethernet devices to reach controller-oriented data through the HMI's protocol bridging -- the HMI acts as the aggregation point on the network.
- No public IP address or port forwarding is required on the factory firewall for remote access.



HMI role in transparent gateway mode

In transparent gateway mode, the industrial HMI acts as a network gateway on the shop floor:

- It maintains native field-bus connections to one or more programmable controllers (serial, Ethernet, or proprietary industrial network).

- It transparently bridges communication so that upstream devices on the Ethernet side can access controller data through the HMI, without each upstream system needing a direct controller connection.
- From Privitty's perspective, the HMI is the single logical endpoint on the LAN for both HMI visualization (VNC) and structured data (OPC UA).

OPC UA server (gateway / aggregator)

The HMI's built-in OPC UA server:

- Runs in server-only mode (it does not itself act as an OPC UA client toward the controller).
- Collects tag data from connected controllers via native field-bus protocols.
- Publishes a unified OPC UA address space representing controller tags, alarms, and device status.
- Listens on the shop-floor network at an endpoint URL of the form opc.tcp://<hmi_ip>:<port>, with the exact port defined by the HMI's own configuration.

Native VNC server

Many industrial HMI platforms provide a built-in VNC server for remote screen viewing and operation, subject to the specific model, firmware, and license options in use.

- VNC listens on the HMI's shop-floor IP address, typically on port 5900.
- The remote operator never connects to the HMI directly over the internet; access is always mediated through Privitty Edge.

Privitty Edge Role

Privitty Edge runs on the gateway device and provides two access paths to a remote operator: an SSH tunnel with local port forwarding for ad-hoc protocol access, and a managed OPC UA data path for continuous structured monitoring.

SSH tunnel with local port forward

Used when the operator prefers SSH as the transport. The operator-local Privitty SSH port (commonly 2222) is never exposed on the gateway's public interface -- it exists only inside the operator's own encrypted Privitty session.

Operator steps

1. Join the gateway via the Privitty App.
2. Open the SSH tunnel by selecting the remote access option in the Privitty App.
3. From a terminal on the operator device, run:

```
ssh -L 5900:<hmi_ip>:5900 -p 2222 <username>@127.0.0.1
```

- 2222 is the operator-local Privitty SSH port, not a port on the gateway's public interface.
- <hmi_ip> is the HMI's address on the shop-floor LAN (for example, 192.168.1.20).
- SSH standard port forwarding (-L) runs inside the encrypted SSH session; Privitty's transport is byte-transparent to it.

4. Launch a VNC viewer and connect to 127.0.0.1:5900 (port 5900 is the default VNC port).

Industrial data collection -- OPC UA path

Sequence of flow:

5. The industrial HMI continuously synchronizes controller tag data via native field-bus protocols.
6. The HMI's OPC UA server exposes tags, alarms, events, and status on the shop-floor network.
7. The Privitty OPC UA client on the gateway (co-located with Privitty Edge) connects to the HMI's opc.tcp endpoint, browses the address space, and subscribes to tag changes via Subscribe/MonitoredItems.
8. Values are processed locally or forwarded through the Privitty Edge HTTP API (privitty-edged, local JSON-RPC port 7200).
9. Remote operators, MES systems, or the Privitty Watchtower dashboard receive data for all monitored controllers.

Data Monitoring

Privitty provides industrial data collection and alerting capabilities, all visible through Privitty Watchtower.

Feature	Privitty capability
OPC UA tag polling / subscription	OPC UA client running on Privitty Edge
Threshold alarms	Supported -- per-tag threshold rules generate alarm events
Native controller alarms (OPC UA Alarms & Conditions, Part 9)	Supported where the OPC UA server implements Part 9 (see Open Items)
Controller diagnostic / fault logs	Supported where configured (see Open Items)
Remote access session logs	Full session detail with admin revoke control, time-based or conditional
AI-enabled log analysis	Watchtower AI can analyze logs, alarms, and events and suggest a remedy
Human alerts and notification	In emergency conditions, AI can notify the relevant person(s) on their Privitty App with a notification and a potential remedy

Threshold alarms

The OPC UA client on Privitty Edge supports per-tag threshold rules, generating alarm events through the Privitty Edge REST API:

```
OPC UA tag read → value > threshold → POST send_msg
```

Native controller alarms

Privitty Edge implements OPC UA Alarms & Conditions (Part 9) subscription in its OPC UA client:

- Subscribes to ConditionType / AlarmConditionType nodes on the HMI's OPC UA server.
- Forwards alarm events -- including severity, active state, and acknowledgement state -- through Privitty Edge.

Whether native controller alarms are reachable this way depends on whether the specific HMI's OPC UA server implements Part 9 and exposes controller alarm queues as OPC UA Condition objects -- this should be confirmed for each target platform.

Log types

- Privitty Edge provides metadata for each session log: chat ID, session ID, tunnel protocol, timestamps, and related fields.
- Gateway system metrics via systemd: CPU, RAM, I/O utilization, and health.

- Privitty Edge–specific utilization parameters and logs.

Privitty Edge -- Resource Utilization

RAM

Component	Typical footprint
Rust binary base (privitty-edged)	~15–20 MB
Tokio async runtime + threads	~2–4 MB
SQLite connection pool (WAL mode)	~8 MB cap (configurable; hardcoded PRAGMA soft_heap_limit = 8388608 in source)
Tunnels (when active)	~4–8 MB per active tunnel
IMAP/SMTP connections	~2–4 MB per account
Total at idle	~30–40 MB
Total during active tunnel	~50–70 MB

CPU

Activity	CPU characterization
Idle	Near zero
Receiving a Privitty PDU / handshake	Brief spike: ChaCha20-Poly1305 + BLAKE3 + X25519, sub-millisecond on a Raspberry Pi 4–class device
Active tunnel (SSH/RDP/VNC byte bridge)	Proportional to throughput; tokio::io::copy is pure I/O forwarding with minimal CPU. Light interactive sessions (SSH) use <1% CPU; RDP screen streaming may reach 5–15%
Tunnel handshake	~0.5–1s of moderate CPU at establishment, then near-zero during data flow
Housekeeping (once per 24h)	Short burst: blob directory scan + SQLite WAL checkpoint, a few seconds
Sustained idle	<1% on a Pi 4–class device
Active tunnel	2–15% on a Pi 4–class device, depending on protocol and activity

Storage

Location	Contents	Characteristics
{accounts}/dc.db	SQLCipher SQLite database -- messages, keys, Privitty metadata, PDU records	Grows with message/file history; 5–50 MB typical

Location	Contents	Characteristics
{accounts}/dc.db-blobs/	Encrypted file blobs and attachments (.prv)	Main variable storage -- each sent/received file lives here until cleaned
{accounts}/dc.db-wal	SQLite write-ahead log	Truncated automatically on WAL checkpoint (24h housekeeping)
{accounts}/privitty.lic, device.uuid, etc.	License and device identity	Tiny, fixed (~1 KB)

I/O

Activity	I/O pattern
Idle	Near zero -- a persistent TCP connection with no disk I/O
Receiving a Privitty PDU / handshake	Small sequential writes to SQLite WAL (a few KB per message)
File send/receive	One sequential write of the .prv blob; one read when the receiver fetches it
File send/receive (>20 MB)	Streaming sequential read from blob directory; one write on the receiver side
Active tunnel	Zero disk I/O -- the tunnel is a pure in-memory QUIC ↔ TCP byte bridge (tokio::io::copy); nothing is written to disk during the session
Housekeeping (daily)	One blob directory scan, deletion of orphan files, and a WAL checkpoint

Secure Operator-to-Factory Program Transfer

Industrial operators increasingly need a secure, enterprise-grade capability for remote program transfer to controllers and HMIs from operator locations outside the factory network. The solution must protect intellectual property and production integrity while enabling authorized engineers to deploy, update, and verify automation projects without on-site presence.

Privitty is proposed as a software stack running on an edge industrial PC at the factory, with operator access from mobile and desktop devices.

Key outcomes

Outcome	How Privitty delivers
Secure program transfer	Encrypted file transport with per-file access control, forward restrictions, and true revoke
Remote engineering access	E2EE tunnels (RDP, VNC) to edge-hosted tools and remote-HMI client software
No cloud data retention	Production programs and session payloads remain on the edge; the relay is transport-only
Enterprise governance	Dedicated Privitty Watchtower to provision devices, operators, and policies
OEM branding	Branded operator app, edge service, Watchtower, and private relay under the partner's control
No additional hardware	Software deployment on existing edge industrial PC / Windows IoT Enterprise hardware

Requirement alignment

Stated requirement: remote program transfer to controllers and HMIs from a remote location.

Need	Privitty capability	Notes
Transfer controller project files to the plant	E2EE encrypted file send to Privitty Edge on the edge PC	Supports project archives, backup files, recipe exports, and customer-defined artefacts
Transfer HMI project files to the plant	Same encrypted file channel	HMI design-tool project packages delivered to the edge
Verify and apply programs on-site	RDP/VNC tunnel to the edge PC	Engineer runs engineering tools on the edge PC or validates via local automation
Operate / monitor HMI remotely	Remote-HMI client + edge PC VNC/RDP via E2EE tunnel	Ready-to-use remote HMI access path already present on the edge PC

Need	Privitty capability	Notes
Authorize who may transfer programs	Watchtower operator provisioning + Privitty pairing	Only registered operators can reach a given edge
Audit remote activity	Watchtower monitoring + edge event stream	Session, transfer, and tunnel visibility for administrators
Keep plant network closed	Outbound-only edge connectivity via private relay	No inbound firewall holes to the controller or HMI
No additional hardware	Software deployment on existing edge industrial PC	Runs on existing Windows IoT Enterprise hardware

Primary vs. supporting capabilities

Primary (proof-of-concept focus): encrypted program file transfer from operator to the edge, with access control and administrative visibility.

Supporting (integrated in the proof of concept, essential to the workflow): E2EE RDP/VNC tunnels to the edge for engineering sessions and remote-HMI access.

Design principles

10. Edge-first -- data and decrypted programs are handled on the edge PC, not in the cloud.
11. End-to-end encryption -- operator-to-edge channels are encrypted; the relay cannot read payloads.
12. Cryptographic identity -- each edge and operator has a verified identity established through secure pairing.
13. Least privilege -- transfer and tunnel rights are granted per operator, per device, and per session.
14. Sovereignty -- the partner operates the private relay and dedicated Watchtower under its own infrastructure and policies.

Architecture overview

Privitty implements a four-part architecture: an edge gateway on the factory floor, a private relay for encrypted signalling, an operator-facing application, and a centralized management console.

Factory floor -- edge industrial PC (Privitty Edge)

The edge industrial PC acts as the secure gateway between the factory LAN and remote operators.

Component	Function
Privitty Edge (privitty-edged)	Long-running Windows service; E2EE gateway, encrypted file I/O, tunnel bridge, local JSON-RPC API
Remote-HMI client software	Vendor-provided remote HMI client/runtime on the edge PC for HMI access
Edge PC VNC / RDP	Native remote desktop services for engineering tool sessions
Local automation (optional)	Scripts or MES/SCADA integration via HTTP JSON-RPC on localhost:7200

Factory LAN connectivity is unchanged by Privitty's presence:

```
Edge PC -- LAN (VPN) → HMI → Controller
```

Privitty terminates encrypted sessions on the edge PC. Controller and HMI programming traffic to field devices remains on the local industrial network, using existing engineering tools and native protocols.

Private relay

A dedicated Privitty relay, operated by the partner organization or a partner-designated hosting environment, provides:

- Encrypted message and file signalling between peers.
- NAT traversal assistance for QUIC peer-to-peer large file transfer.
- No storage of message bodies, file content, or session metadata beyond transient transport.

This relay is not shared with the public Privitty cloud. Hostname, certificates, and network policies remain under the partner's control.

Remote operator -- Privitty App (white-label)

The branded operator application runs on mobile and desktop platforms.

Capability	Use in the workflow
Encrypted file transfer	Send controller/HMI project packages to the edge
Access control UI	Allow download, restrict forward, set expiry, revoke access
RDP tunnel	Full engineering desktop on the edge PC
VNC tunnel	Lightweight remote session / remote-HMI-adjacent workflows
Secure pairing	QR code / invite link to authorized edges only

Privitty Watchtower

A separate Watchtower deployment, dedicated to the partner organization, provides the web-based management console.

Function	Description
Device registry	Onboard edge industrial PC / Privitty Edge units (factory assets)
Operator registry	Provision mobile and desktop operators by role
Activity monitoring	View pairing status, file transfers, active tunnels
Policy and provisioning	Assign operators to edges, manage licenses, trigger revoke
Fleet visibility	Multi-site dashboard for organization and end-customer administrators

Watchtower manages identity and governance. It does not store production program content.

Primary Workflow -- Remote Program Transfer

This section describes the priority use case: delivering controller and HMI programs from a remote operator to equipment on the factory floor.

Actors

- Administrator -- provisions edges and operators in Watchtower.
- Remote engineer (operator) -- authorized to transfer programs to a specific edge.
- Privitty Edge -- receives, decrypts, and exposes programs per access policy.
- Local engineering toolchain -- applies programs to the controller and HMI over the LAN.

Step-by-step description

15. Onboarding -- the administrator logs in to the Privitty Watchtower management console.
16. Provisioning the edge -- the Privitty Edge service is installed on the edge PC and started (sc start privitty-edge); a license activates the edge and generates its device identity, displayed as a pairing QR code.
17. Adding the edge and operators -- the administrator scans the edge's QR code into Watchtower and emails the license to the operator(s); on enrollment, operators are automatically added to the Watchtower roster.
18. Pairing and handshake -- the administrator pairs the edge to specific operators and sets a time duration; the operator scans the edge's QR code in the Privitty App, which establishes the end-to-end encrypted channel and access-controlled sharing capabilities (SSH, RDP, file transfer).
19. Remote session -- the operator opens a tunnel from the edge bridge to the target HMI service; SSH/RDP/VNC traffic flows over the E2EE tunnel. The administrator retains the ability to revoke access at any time from Watchtower.
20. File transfer -- Privitty Edge sends an encrypted file to the operator with defined attributes (for example: a 30-minute time duration, download disabled, forwarding disabled); the operator can only view the file in the Privitty App's secure viewer under those constraints.
21. Monitoring -- the administrator monitors transfer and session activity from Watchtower throughout.
22. Revocation -- the operator (or administrator) revokes file access once the engagement is complete.

File transfer characteristics

- Encryption -- all program packages are encrypted end-to-end before leaving the operator device.

- Access control -- per-file policies: allow download, deny forward, time-bound access, and true revoke (access can be withdrawn after delivery).
- Large projects -- QUIC peer-to-peer transfer when both peers are online, with relay fallback for signalling.
- Automation hook -- the edge exposes JSON-RPC and server-sent events on localhost:7200 so integrators can auto-ingest decrypted packages into a controlled directory or workflow.

What Privitty does not do

- Privitty does not interpret vendor-specific engineering file formats.
- Privitty does not write directly to controller memory -- program application remains with vendor engineering tools on the edge PC or the factory LAN.
- Privitty does not store programs on the relay or in Watchtower.

This separation preserves the engineering toolchain's authority and keeps the security boundary clear.

Supporting Workflows -- HMI Remote Access

Remote program transfer is often accompanied by verification, commissioning, and operator guidance on the HMI. The edge PC already provides the building blocks for this:

Capability	Privitty integration
Remote-HMI client software	Engineer reaches the edge PC via E2EE RDP/VNC tunnel; the client provides ready-to-use HMI remote access on the factory LAN
Edge PC VNC	Privitty VNC tunnel → edge PC local VNC service → engineering or monitoring session
Edge PC RDP	Privitty RDP tunnel → full Windows desktop for engineering tools and remote-HMI client software

Typical combined session

23. Engineer pairs with the edge.
24. Engineer sends the HMI/controller program package (encrypted file transfer).
25. Engineer opens an RDP tunnel to the edge PC.
26. On the edge PC desktop: open the engineering tool and apply the program to the controller/HMI.
27. Launch the remote-HMI client to verify HMI screens remotely.
28. Close the tunnel; revoke file access; the session is logged in Watchtower.

Remote access tunnel summary

Protocol	Default operator shim port	Typical edge PC target	Use case
RDP	3389	Edge PC RDP service	Full engineering desktop
VNC	5900	Edge PC VNC service	Lightweight remote GUI
SSH	2222	OpenSSH on the edge PC (if enabled)	Scripting / automation

Tunnels are operator-initiated only, with one active tunnel per chat session in the current release, and use E2EE iroh QUIC transport between peers.

Security and Compliance Posture

Security layers

Layer	Mechanism
Identity & pairing	Securejoin with QR / invite; OpenPGP-verified peers
Data in transit	End-to-end encryption for all Privitty messages and files
Remote access	E2EE tunnel; no cleartext protocol exposure over the internet
Access control	Per-file download / forward / expiry / revoke
Relay	Partner-private; transport-only; no payload retention
Edge binding	License and device identity bound to the specific edge instance
Governance	Watchtower provisioning, monitoring, and administrative revoke
Network exposure	Outbound-only from the edge; no inbound connections to the plant VLAN

Data residency

Data type	Location
Controller/HMI program files (decrypted)	Edge storage only
Encrypted payloads in transit	Ephemeral on relay
Administrative metadata	Partner's Watchtower instance (policy-defined retention)
Cryptographic keys	Operator device and edge instance

Security benefits for OEM / white-label partners

- Brand trust -- the partner controls the relay, Watchtower, and operator distribution.
- Customer isolation -- a private relay prevents cross-tenant transport.
- Auditability -- central visibility without central storage of program intellectual property.
- Revocation -- immediate withdrawal of file access and session teardown.

OEM Deployment Model

Privitty is designed to be offered as a partner OEM software capability, integrated into a partner's solution stack for edge-PC–based remote service.

Deployment components

Component	Deployment treatment
Privitty Edge	Pre-installed or partner-delivered installer for the edge PC / Windows IoT Enterprise
Operator App	Partner-branded mobile and desktop application
Watchtower	Dedicated partner instance (hosting per partner cloud policy)
Relay	Partner-private relay cluster
Documentation	Partner-branded deployment, operator, and security guides

Edge PC service deployment

Privitty Edge runs as a Windows service on the edge PC:

```
sc start privitty-edge
```

Parameter	Purpose
--server	Partner-private relay hostname
--profile	Human-readable edge display name
--listen	Local JSON-RPC API (default 127.0.0.1:7200)
Tunnel env vars	Map RDP/VNC targets to edge PC local services

Licensing

- License seats are managed through Watchtower and edge activation.
- Device identity is generated on first activation and bound to the specific edge instance.
- Operator licenses are provisioned per engineer or per customer contract.

End-customer experience

End customers (factories) receive:

29. An edge PC with Privitty Edge pre-configured for their private relay tenant.
30. The operator app, distributed via partner channels.
31. Watchtower access for their administrators, role-scoped by the partner.
32. No requirement for additional Privitty hardware.

Demonstration Environment (Lab Equivalent)

For integration testing without physical industrial hardware, a cloud VPC can simulate the shop-floor subnet.

Lab instance	Simulates
Gateway	Gateway + Privitty Edge + OPC UA client
HMI	Industrial HMI (VNC + OPC UA server)
Controller	Reference OPC UA source; not used by Privitty Edge directly

The lab validates:

- Gateway → HMI OPC UA connectivity, using a reference OPC UA server demo on the standard endpoint and port.
- Gateway → HMI VNC via SSH port forward.
- Privitty Edge tunnel configuration patterns.

Production deployment replaces the demo servers with the native HMI VNC and OPC UA server.

Proof of Concept Plan

Objective

Demonstrate remote program transfer to a controller and HMI from a remote operator to a factory-edge industrial PC, with encrypted delivery, controlled access, RDP-based engineering access, and Watchtower visibility.

Proof-of-concept environment

Item	Specification
Edge hardware	Any Windows IoT Enterprise industrial PC or equivalent
Relay	Partner-private relay (staging instance)
Watchtower	Partner dedicated staging instance
Operator devices	One mobile + one desktop
Field equipment	Industrial HMI and controller (or lab equivalents)
Engineering software	Remote-HMI client, vendor engineering / design tools (as applicable)

Success criteria

#	Criterion	Verification
1	Administrator registers edge and operator in Watchtower	UI + API confirmation
2	Operator pairs with edge over private relay	E2EE established
3	Operator sends controller program package to edge	Encrypted transfer completes
4	Operator sends HMI program package to edge	Encrypted transfer completes
5	Access policy enforced (no forward, revoke works)	Negative test + revoke test
6	Engineer opens RDP session to edge PC via Privitty	Desktop reachable
7	Program applied to controller/HMI using engineering tools	Field device updated
8	Remote-HMI client used to verify HMI	Remote HMI operation confirmed
9	Watchtower shows transfer and session activity	Audit log review
10	Relay confirmed transport-only	No payload retention audit

Indicative timeline

Week	Activity
1–2	Staging relay + Watchtower; edge install on Windows IoT Enterprise
3	Pairing, file transfer, access control validation
4	RDP tunnel + remote-HMI client + program apply workflow
5	Security review, documentation, stakeholder demo

Implementation Roadmap

Phase 1 -- Proof of concept

- Partner-private relay (staging)
- Dedicated Watchtower (staging)
- Privitty Edge on Windows IoT Enterprise
- Branded operator app (beta)
- Remote program transfer + RDP + remote-HMI client validation

Phase 2 -- Pilot (field trial)

- Multi-site edge onboarding
- Role-based operator management in Watchtower
- Integration hooks (JSON-RPC ingest path for program packages)
- Hardened installer for the edge PC manufacturing image
- Customer administrator training materials

Phase 3 -- Production release

- Production relay cluster (high availability)
- Production Watchtower with partner SSO (if required)
- Edge PC factory image integration
- App store / partner download distribution
- Security certification package and customer deployment playbooks

Deliverables

From Privitty

Deliverable	Description
Privitty Edge for Windows IoT Enterprise	Service binary, installer, partner branding
Operator application	Mobile and desktop, partner branded
Watchtower deployment package	Dedicated instance setup and configuration guide
Private relay deployment package	High-availability topology, certificates, network requirements
Integration guide	Edge PC image integration, tunnel port mapping, remote-HMI workflow
Security whitepaper	Architecture, encryption, data flows, threat model
API reference	JSON-RPC for automation on the edge
Proof-of-concept test plan	Reproducible validation script

Joint deliverables

Deliverable	Typical owner
Controller/HMI program transfer runbook	Partner + Privitty
Watchtower operator provisioning SOP	Partner
Customer-facing product datasheet	Partner
Detailed event-flow architecture diagram	Partner (with Privitty review)

Assumptions and Dependencies

Assumptions

33. Remote program transfer is initiated by authorized engineers, not unattended public endpoints.
34. Program application to the controller/HMI uses existing vendor engineering tools on or via the edge PC.
35. The edge PC has outbound connectivity to the private relay (HTTPS, IMAP/SMTP, UDP for QUIC).
36. Remote-HMI client software and edge PC VNC/RDP are installed and licensed per standard vendor practice.
37. The partner organization distributes the branded operator app to end customers.

Dependencies

Dependency	Required for
Staging/production hosting for private relay	All remote connectivity
Watchtower hosting decision	Fleet management
Windows IoT Enterprise proof-of-concept hardware	Proof-of-concept execution
Sample controller/HMI project packages	Transfer validation
Remote-HMI client + engineering tool licensing in lab	End-to-end apply workflow
Partner security review inputs	Production release
Partner branding guidelines	App and documentation

Open items for joint review

- OPC UA endpoint URL and port -- default port and security policy (None / Sign / SignAndEncrypt).
- OPC UA address space -- tag naming, namespace URI, and subscription limits.
- Native controller alarms (OPC UA Alarms & Conditions, Part 9) -- whether the target OPC UA server implements Part 9 and exposes controller alarm queues as Condition objects.
- VNC authentication mode, simultaneous sessions, and whether VNC can be restricted by client IP.
- Read/write policy -- which OPC UA access levels are exposed to external clients.

- Certification -- any constraints on third-party software (Privitty Edge, OPC UA client) coexisting on the same LAN segment.
- Exact program file types and maximum package sizes per customer segment.
- Automated ingest path vs. manual engineer apply (RDP) as the default workflow.
- Watchtower single sign-on integration with the partner's identity provider.
- End-customer vs. partner-administered Watchtower tenancy model.
- Regulatory documentation requirements, such as IEC 62443 mapping.

Appendix A -- Event Flow Summary

Phase	Event	Initiator	Result
Provisioning	Edge registered in Watchtower	Administrator	Edge identity created
Provisioning	Operator account created	Administrator	Operator identity created
Activation	Edge license activated on the edge PC	Edge / Admin	Edge joins the fleet
Pairing	Operator scans edge QR / invite	Operator	Securejoin started
Pairing	Securejoin completes	Edge + Operator	Verified peer relationship
Handshake	Privitty E2EE established	Edge + Operator	Encrypted channel ready
Handshake	File access control negotiated	Edge + Operator	Transfer policy context set
Discovery	Edge publishes capabilities (RDP/VNC/file)	Edge	Operator UI shows available actions
Transfer	Operator sends controller/HMI program package	Operator	Encrypted package delivered to edge
Transfer	Edge decrypts and stages file	Edge	Program available per access policy
Access	Operator opens RDP tunnel	Operator	E2EE session to edge PC desktop
Apply	Engineering tools write program to controller/HMI	Engineer	Field devices updated
Verify	Remote-HMI client check	Engineer	Commissioning validation
Governance	Watchtower records activity	System	Audit entry created
Teardown	Operator revokes file access	Operator	Access withdrawn
Teardown	Tunnel closed	Operator	Session ended

Appendix B -- Component Reference

Name	Role	Typical deployment
Privitty Edge	Factory gateway daemon	Windows service on the edge PC
Privitty App	Operator client	Partner-branded mobile / desktop
Privitty Relay	Encrypted transport	Partner-private cluster
Privitty Watchtower	Fleet management web UI	Partner dedicated instance
Remote-HMI client software	Remote HMI access	Pre-installed on the edge PC
Edge PC VNC/RDP	Remote desktop	Native Windows services
Industrial HMI	Human-machine interface	Factory LAN
Programmable controller	Process control	Factory LAN

Summary

Privitty enables secure remote program transfer to controllers and HMIs, secure remote HMI visualization, and structured OPC UA data integration as an edge-deployed capability -- without exposing factory networks or storing production intellectual property in the cloud. Encrypted file transfer addresses the core remote-engineering requirement; E2EE RDP/VNC tunnels and OPC UA data integration complete the engineering and monitoring workflow. A private relay and dedicated Watchtower instance keep transport and governance under the deploying organization's control.

The recommended next step is a proof of concept on Windows IoT Enterprise (or an equivalent industrial PC) to validate program transfer, access control, RDP engineering access, OPC UA data monitoring, and Watchtower audit visibility -- followed by pilot deployment with field integration teams.

“Machine data flows freely -- access never does”