Oracle Hospitality Payment Interface and Token Proxy Service

Deployment Guide

Release 20.3
F35687-02
February 2024

Oracle Hospitality Payment Interface and Token Proxy Service Deployment Guide Release 20.3

F35687-02

Contents

Contents 3

Preface 4

1 OPI and TPS Overview 1-1

2 OPI Deployment 2-1

Installing All-In-One 2-1
Installing Application Components and Database on Separate Servers 2-2

3 TPS Deployment 3-1

Installing All-In-One 3-1
Installing Application Components and Database on Separate Servers 3-3

4 Customer Deployment Scenarios 4-1

OPERA On-Premise Single Property 4-1
Self-Hosted Multi-Property 4-2
Oracle-Cloud Hosted Multi-Property 4-5
Suite8 Chip and Pin OPI 4-6

Preface

Purpose

This document describes how to organize environments for an installation of the Oracle Payment Interface (OPI) across OPERA Property Management System (PMS) Single Property, OPERA PMS Multi-Property, OPERA V5 SAAS (Oracle Hosted) and OPERA Cloud. Using this guide, you can determine the number of machines and OPI instances needed to process both financial and token transactions.

This document also describes how you can install the Token Proxy Service (TPS) service, web service, configuration web portal and database on one server or on separate servers that means customer can have one server for DB - Oracle or MySQL, one server for web portal - WebLogic or Tomcat and multiple TPS servers for self-hosted TPS.

Audience

This document is intended for customers who need to deploy OPI and TPS.

Customer Support

To contact Oracle Customer Support, access the Customer Support Portal at the following URL:

https://iccp.custhelp.com

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

Documentation

Oracle Hospitality product documentation is available on the Oracle Help Center at

http://docs.oracle.com/en/industries/hospitality/

Table 1-1 Revision History

Date	Description
July 2020	• Initial Publication
November 2020	• Updated for release 20.3
April 2023	• Updated communication diagrams in Customer Deployment Scenarios chapter

Date	Description
May 2023	- Updated the Customer Support section with the new support portal name and URL
October 2023	- Removed the MySQL 5.6 version across the document as the OPI installer no longer supports
February 2024	- Added Windows 2022 in OPI Deployment and TPS Deployment chapters

1

OPI and TPS Overview

Oracle Payment Interface (OPI) is a payment card-processing interface that integrates with the OPERA PMS Single Property, OPERA PMS Multi-Property, OPERA V5 SAAS (Oracle Hosted) and OPERA Cloud, Suite 8 (PMS) and SPMS (Cruise). It defines a standard interface that partner payment service providers can implement to provide credit card processing functionality to Oracle Hospitality OPERA, Suite 8 (PMS) and SPMS (Cruise).

The Token Proxy Service is designed to provide token exchange proxy service for hosted applications. It is a proxy interface that works with your application to communicate with payment service providers (PSP), on whom it relies to provide the actual token functionality. It connects to PSPs via internet or virtual private network (VPN). The firewall should be configured to allow connection from OPERA to TPS and from TPS to PSP for token exchange.

2

OPI Deployment

For hotel deployment, OPI can be installed on existing machine which has sufficient resources to accommodate OPI.

The following lists the minimum specification for OPI:

- **Processor**: Dual core and 64bit
- **Available Disk space**: 6 GB free minimum
- **Available memory**: 4 GB RAM minimum

Installing All-In-One

The Oracle Payment Interface release 20.3 is compatible with the following operating systems:

  - Microsoft Windows 10 Professional
  - Microsoft Windows 10 Enterprise
  - Microsoft Windows 11 Professional
  - Microsoft Windows 11 Enterprise
  - Microsoft Windows Server 2012 R2
  - Microsoft Windows Server 2016
  - Microsoft Windows Server 2019
  - Microsoft Windows Server 2022

With an all-in-one installation, you can install the OPI service, OPI utility service, OPI configuration tool and database on one server.

OPI Config Service

Deals with connections from applications used to configure OPI, such as OPI configuration Tool and Wizard.

OPI Service

- The OPI Service is the main OPI Application service, listening for connections to OPI from PMS and making connections to PSP.
- Always restart the OPI Service after creating or changing any configuration.

OPI Utility Service

The OPI Utility Service handles any configuration values that are encrypted, such as passwords and passphrases.

Database

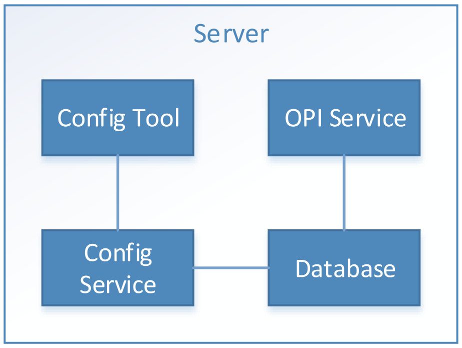
The Oracle Payment Interface Installer release 20.3 supports the following database connections:

- MySQL Database 5.7 and 8.0
- Oracle Database 11g / 12c / 19c

OPI Config Tool

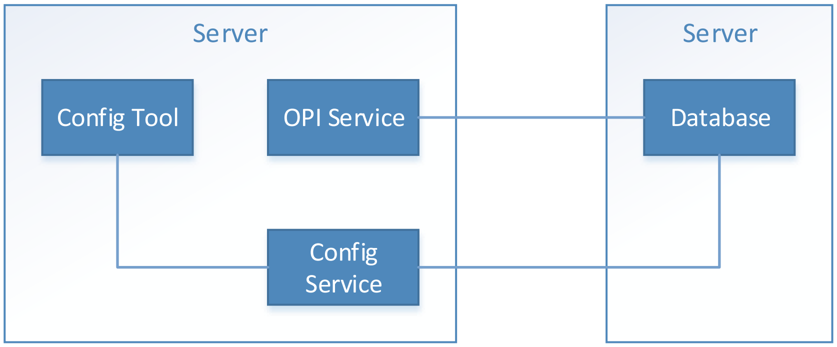
The Configuration Tool includes a configuration wizard and full configuration tool.

- **LaunchWizard.bat** – contains the most used settings that should be sufficient to allow configuration of a basic working merchant configuration.
- **LaunchConfiguration.bat** – contains some additional advanced settings that may be required in certain installations.



Installing Application Components and Database on Separate Servers

You can install OPI service, OPI utility service, and OPI configuration tool on one physical server and install the database on a separate server.



3
TPS Deployment

Installing All-In-One

The Oracle Payment Interface release 20.3 is compatible with the following operating systems:

- Microsoft Windows 10 Professional
- Microsoft Windows 10 Enterprise
- Microsoft Windows 11 Professional
- Microsoft Windows 11 Enterprise
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2016
- Microsoft Windows Server 2019
- Microsoft Windows Server 2022

With an all-in-one installation, you can install the TPS service, web service, configuration web portal and database on one server.

Components

Token Proxy Service has three main components:

- Database
- Token Proxy Web Portal
- Token Proxy Service

The Database

- Supports Oracle 11g /12c / 19c database.
- The database stores the configuration and audit log from the Web Portal.
- The Token Proxy Service requires read-only access to the database. If required, you can configure a different Token Proxy Service database user with less privileges.
- The Web Portal requires a database user with privileges to make changes within the Database.

The Web Portal

- The Web Portal is used to configure the settings used by the Token Proxy Service.

- The Web Portal is a web application supplied in a WAR file suitable for hosting in WebLogic or Tomcat. It relies on the selected web server to store some of its configuration, such as the database connection string (datasource), and to provide a trusted SSL certificate for connections from users accessing the configuration web portal.

Token Proxy Service

The Token Proxy standalone application runs automatically as a service.

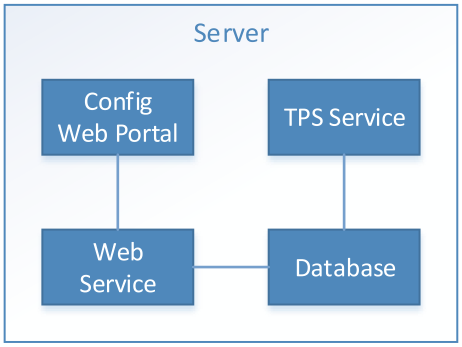
Connections

Connections from OPERA

- The application creates a Listener to monitor a TCP port for XML messages posted over HTTPS. The default Listener port is 443, but it can be set to a custom port number via the Token Proxy Web Portal. This Listener must be exposed to the client (for example, OPERA systems).
- The Listener manages its own use of the certificates (TLS1.2) provided by the data center team, so a firewall or load balancer (if present) must not offer any form of HTTPS-to-HTTP bridging functionality. Instead, the connection must be passed directly to the Token Proxy Service.
- The certificates provided must be installed on all servers running the Token Proxy Service in the event the service is installed on multiple machines for load balance or failover. In case the certificate is deployed on the load balancer, a certificate should also be deployed on the TPS app server to establish HTTPS connection from the load balancer to the TPS server. It is highly recommended to use CA signed certificates.

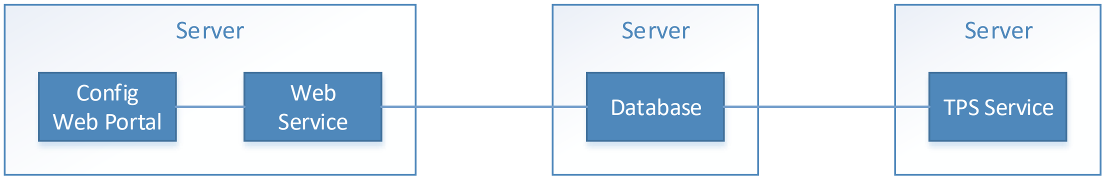
Connections to PSPs (Payment Service Providers)

- The service also makes outgoing connections to PSPs.
- The outgoing connection is to a URL specified by the PSP and the host or port (and optionally a path) is specified by the PSP.
- The outgoing connection can be over the internet or over VPN, but it must use HTTPS with TLS1.2 (but not higher).



Installing Application Components and Database on Separate Servers

You can install TPS service, web service and configuration web portal and database on separate servers.

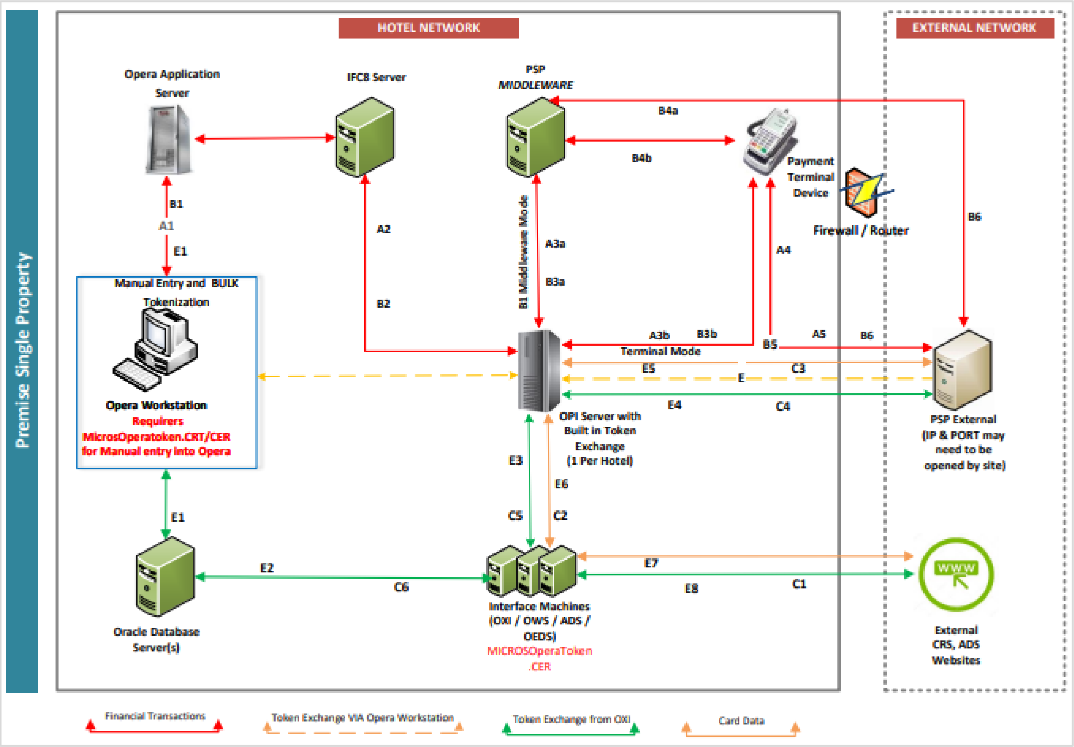


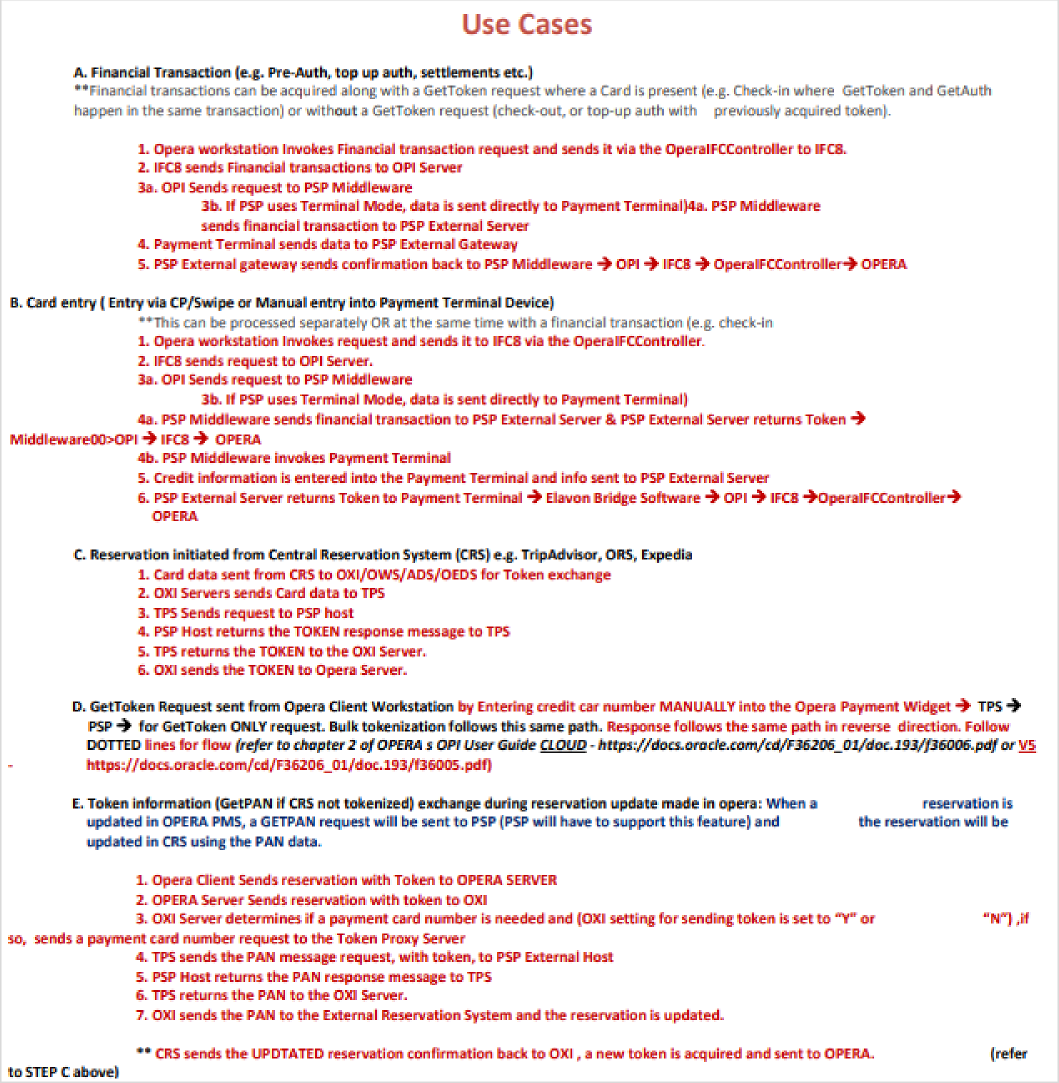
4

Customer Deployment Scenarios

OPERA On-Premise Single Property

- OPERA application components and database installed in hotel network, OPI will be installed in the same network.
- Install one OPI instance to process both financial transaction and token transaction.
- OXI/OEDS will communicate with OPI.
- IFC8 will be used for communication between OPERA and OPI.





Use Cases

A. Financial Transaction (e.g. Pre-Auth, top up auth, settlements etc.)
**Financial transactions can be acquired along with a GetToken request where a Card is present (e.g. Check-in where GetToken and GetAuth happen in the same transaction) or without a GetToken request (check-out, or top-up auth with previously acquired token).

1. Opera workstation Invokes Financial transaction request and sends it via the OperaIFCController to IFC8.
2. IFC8 sends Financial transactions to OPI Server
3a. OPI Sends request to PSP Middleware
3b. If PSP uses Terminal Mode, data is sent directly to Payment Terminal)4a. PSP Middleware sends financial transaction to PSP External Server
4. Payment Terminal sends data to PSP External Gateway
5. PSP External gateway sends confirmation back to PSP Middleware → OPI → IFC8 → OperaIFCController→ OPERA

B. Card entry (Entry via CP/Swipe or Manual entry into Payment Terminal Device)
**This can be processed separately OR at the same time with a financial transaction (e.g. check-in
1. Opera workstation Invokes request and sends it to IFC8 via the OperaIFCController.
2. IFC8 sends request to OPI Server.
3a. OPI Sends request to PSP Middleware
3b. If PSP uses Terminal Mode, data is sent directly to Payment Terminal)
4a. PSP Middleware sends financial transaction to PSP External Server & PSP External Server returns Token → Middleware00>OPI → IFC8 → OPERA
4b. PSP Middleware invokes Payment Terminal
5. Credit information is entered into the Payment Terminal and info sent to PSP External Server
6. PSP External Server returns Token to Payment Terminal → Elavon Bridge Software → OPI → IFC8 →OperaIFCController→ OPERA

C. Reservation initiated from Central Reservation System (CRS) e.g. TripAdvisor, ORS, Expedia
1. Card data sent from CRS to OXI/OWS/ADS/OEDS for Token exchange
2. OXI Servers sends Card data to TPS
3. TPS Sends request to PSP host
4. PSP Host returns the TOKEN response message to TPS
5. TPS returns the TOKEN to the OXI Server.
6. OXI sends the TOKEN to Opera Server.

D. GetToken Request sent from Opera Client Workstation by Entering credit car number MANUALLY into the Opera Payment Widget → TPS → PSP → for GetToken ONLY request. Bulk tokenization follows this same path. Response follows the same path in reverse direction. Follow DOTTED lines for flow *(refer to chapter 2 of OPERA s OPI User Guide CLOUD - https://docs.oracle.com/cd/F36206_01/doc.193/f36006.pdf or* V5 - https://docs.oracle.com/cd/F36206_01/doc.193/f36005.pdf)

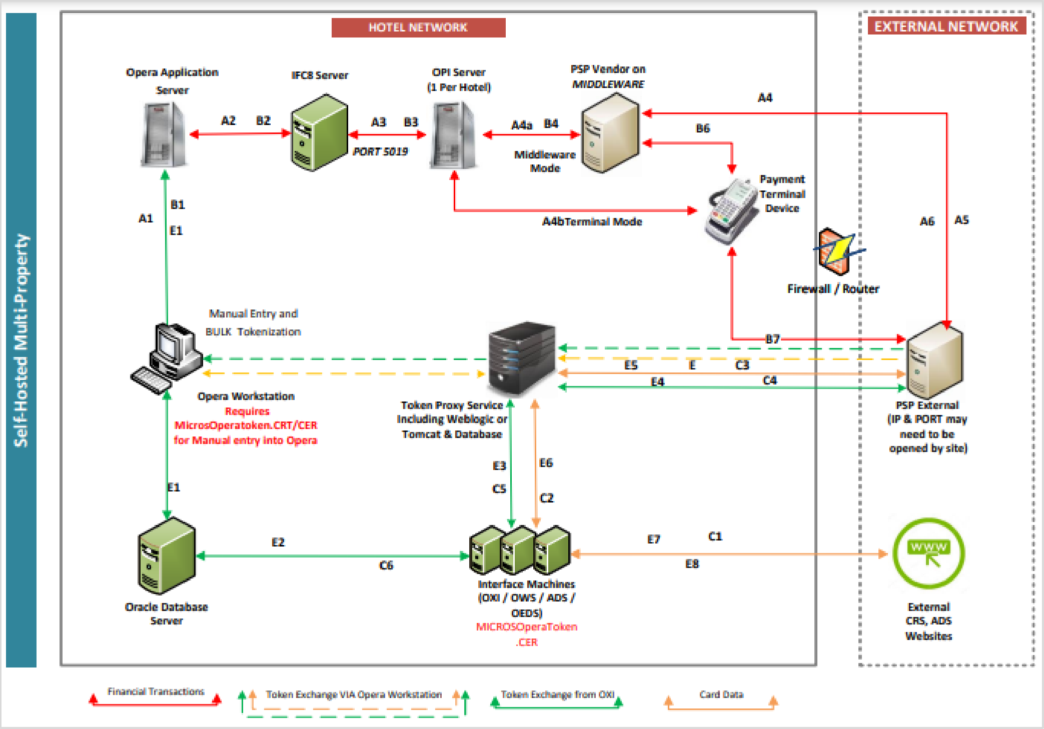
E. Token information (GetPAN if CRS not tokenized) exchange during reservation update made in opera: When a reservation is updated in OPERA PMS, a GETPAN request will be sent to PSP (PSP will have to support this feature) and the reservation will be updated in CRS using the PAN data.

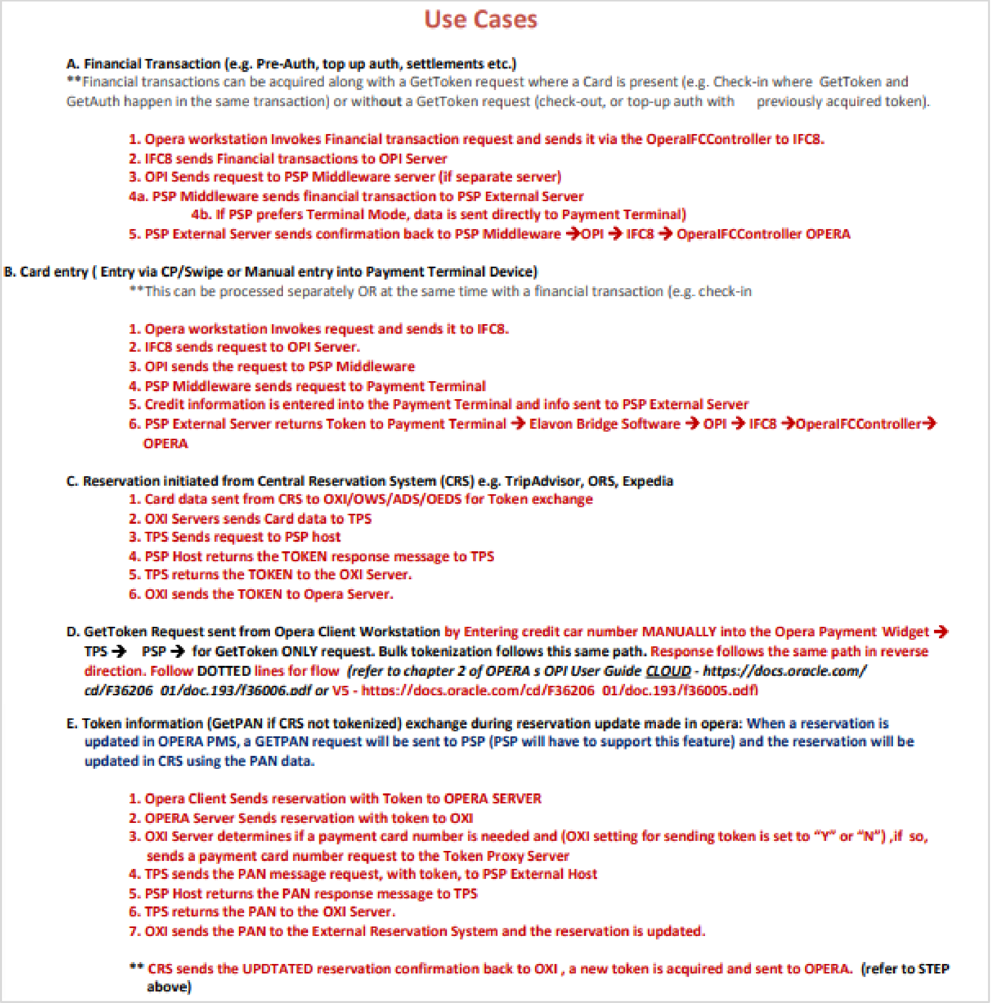
1. Opera Client Sends reservation with Token to OPERA SERVER
2. OPERA Server Sends reservation with token to OXI
3. OXI Server determines if a payment card number is needed and (OXI setting for sending token is set to "Y" or "N") ,if so, sends a payment card number request to the Token Proxy Server
4. TPS sends the PAN message request, with token, to PSP External Host
5. PSP Host returns the PAN response message to TPS
6. TPS returns the PAN to the OXI Server.
7. OXI sends the PAN to the External Reservation System and the reservation is updated.

** CRS sends the UPDTATED reservation confirmation back to OXI , a new token is acquired and sent to OPERA. (refer to STEP C above)

Self-Hosted Multi-Property

- OPERA application components and database installed in hotel network.
- Install one OPI instance per property.
- Install TPS to serve token exchange requests for all the properties.
- OXI/OEDS will communicate with TPS.
- IFC8 (need to be installed unless it is currently in place) will be used for communication between OPERA and OPI.


Self-Hosted Multi-Property
HOTEL NETWORK
EXTERNAL NETWORK
Opera Application Server
IFC8 Server
OPI Server (1 Per Hotel)
PSP Vendor on *MIDDLEWARE*
A2
B2
A3
B3
PORT 5019
A4a
B4
Middleware Mode
A4
B6
Payment Terminal Device
A4bTerminal Mode
A1
B1
E1
A6
A5
Firewall / Router
Manual Entry and BULK Tokenization
B7
E5
E
C3
E4
C4
Opera Workstation
Requires MicrosOperatoken.CRT/CER for Manual entry into Opera
Token Proxy Service Including Weblogic or Tomcat & Database
PSP External (IP & PORT may need to be opened by site)
E3
C5
E6
C2
E1
E2
C6
E7
C1
E8
Oracle Database Server
Interface Machines (OXI / OWS / ADS / OEDS)
MICROSOperaToken .CER
External CRS, ADS Websites
Financial Transactions
Token Exchange VIA Opera Workstation
Token Exchange from OXI
Card Data



Use Cases

A. Financial Transaction (e.g. Pre-Auth, top up auth, settlements etc.)
Financial transactions can be acquired along with a GetToken request where a Card is present (e.g. Check-in where GetToken and GetAuth happen in the same transaction) or without** a GetToken request (check-out, or top-up auth with previously acquired token).

1. Opera workstation Invokes Financial transaction request and sends it via the OperaIFCController to IFC8.
2. IFC8 sends Financial transactions to OPI Server
3. OPI Sends request to PSP Middleware server (if separate server)

4a. PSP Middleware sends financial transaction to PSP External Server

4b. If PSP prefers Terminal Mode, data is sent directly to Payment Terminal)

5. PSP External Server sends confirmation back to PSP Middleware →OPI → IFC8 → OperaIFCController OPERA

B. Card entry (Entry via CP/Swipe or Manual entry into Payment Terminal Device)
**This can be processed separately OR at the same time with a financial transaction (e.g. check-in

1. Opera workstation Invokes request and sends it to IFC8.
2. IFC8 sends request to OPI Server.
3. OPI sends the request to PSP Middleware
4. PSP Middleware sends request to Payment Terminal
5. Credit information is entered into the Payment Terminal and info sent to PSP External Server
6. PSP External Server returns Token to Payment Terminal → Elavon Bridge Software → OPI → IFC8 →OperaIFCController→ OPERA

C. Reservation initiated from Central Reservation System (CRS) e.g. TripAdvisor, ORS, Expedia

1. Card data sent from CRS to OXI/OWS/ADS/OEDS for Token exchange
2. OXI Servers sends Card data to TPS
3. TPS Sends request to PSP host
4. PSP Host returns the TOKEN response message to TPS
5. TPS returns the TOKEN to the OXI Server.
6. OXI sends the TOKEN to Opera Server.

D. GetToken Request sent from Opera Client Workstation by Entering credit car number MANUALLY into the Opera Payment Widget → **TPS → PSP → for GetToken ONLY request. Bulk tokenization follows this same path.** Response follows the same path in reverse direction. Follow **DOTTED** lines for flow *(refer to chapter 2 of OPERA s OPI User Guide CLOUD - https://docs.oracle.com/cd/F36206 01/doc.193/f36006.pdf or V5 - https://docs.oracle.com/cd/F36206 01/doc.193/f36005.pdf)*

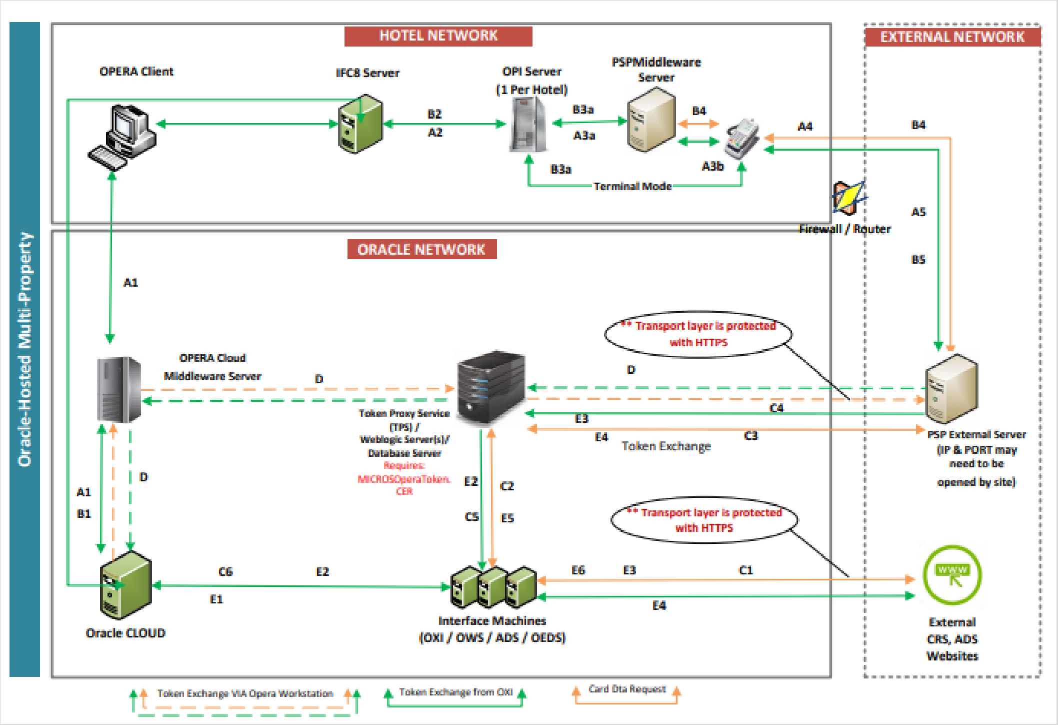
E. Token information (GetPAN if CRS not tokenized) exchange during reservation update made in opera: When a reservation is updated in OPERA PMS, a GETPAN request will be sent to PSP (PSP will have to support this feature) and the reservation will be updated in CRS using the PAN data.

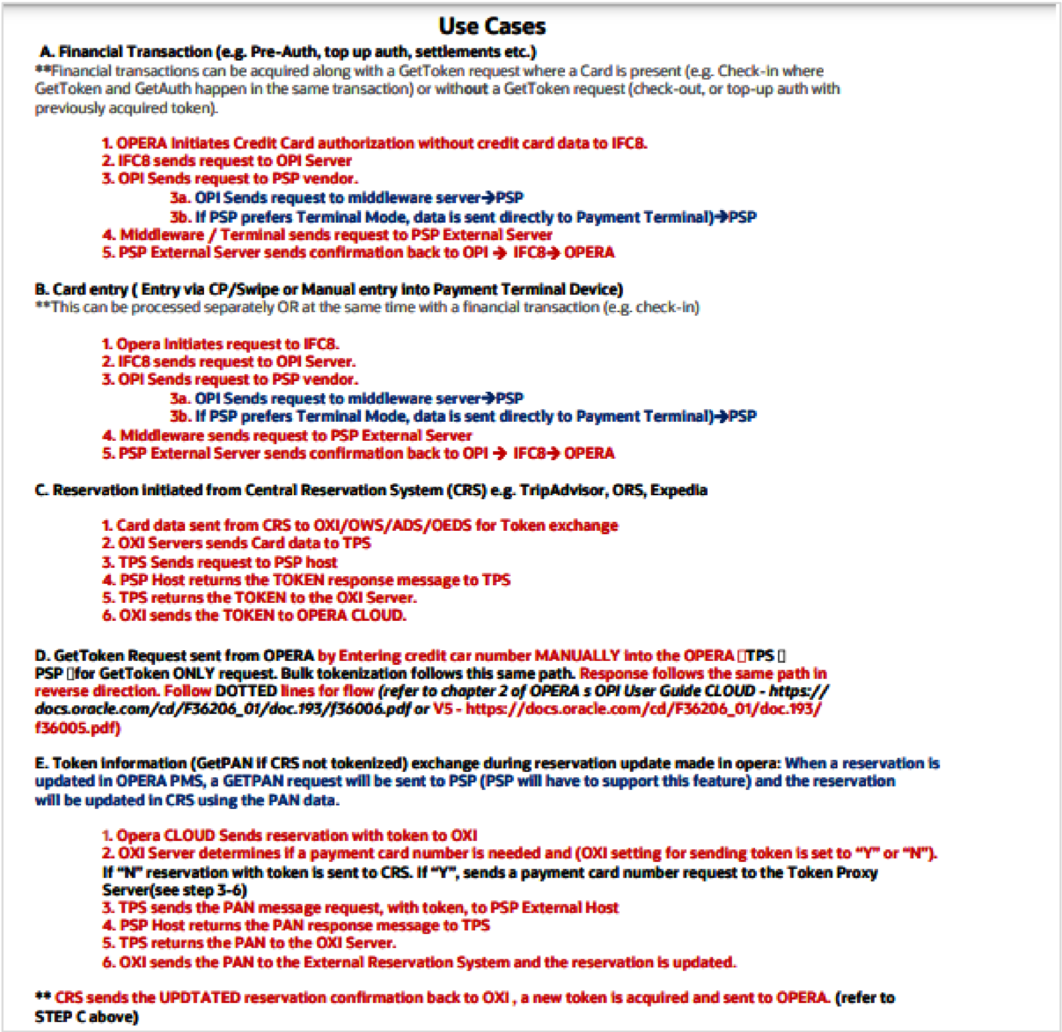
1. Opera Client Sends reservation with Token to OPERA SERVER
2. OPERA Server Sends reservation with token to OXI
3. OXI Server determines if a payment card number is needed and (OXI setting for sending token is set to "Y" or "N") ,if so, sends a payment card number request to the Token Proxy Server
4. TPS sends the PAN message request, with token, to PSP External Host
5. PSP Host returns the PAN response message to TPS
6. TPS returns the PAN to the OXI Server.
7. OXI sends the PAN to the External Reservation System and the reservation is updated.

** CRS sends the UPDTATED reservation confirmation back to OXI , a new token is acquired and sent to OPERA. (refer to STEP above)

Oracle-Cloud Hosted Multi-Property

- Token Proxy Service is deployed in Oracle Datacenter.
- OPERA application components and database installed in Oracle Datacenter.
- Install one OPI instance per property.
- OXI/OEDS will communicate with TPS.
- IFC8 (on property) will be used for communication between OPERA and OPI.





Use Cases

A. Financial Transaction (e.g. Pre-Auth, top up auth, settlements etc.)

**Financial transactions can be acquired along with a GetToken request where a Card is present (e.g. Check-in where GetToken and GetAuth happen in the same transaction) or without a GetToken request (check-out, or top-up auth with previously acquired token).

1. OPERA Initiates Credit Card authorization without credit card data to IFC8.
2. IFC8 sends request to OPI Server
3. OPI Sends request to PSP vendor.
3a. OPI Sends request to middleware server→PSP
3b. If PSP prefers Terminal Mode, data is sent directly to Payment Terminal)→PSP
4. Middleware / Terminal sends request to PSP External Server
5. PSP External Server sends confirmation back to OPI → IFC8→ OPERA

B. Card entry (Entry via CP/Swipe or Manual entry into Payment Terminal Device)

**This can be processed separately OR at the same time with a financial transaction (e.g. check-in)

1. Opera Initiates request to IFC8.
2. IFC8 sends request to OPI Server.
3. OPI Sends request to PSP vendor.
3a. OPI Sends request to middleware server→PSP
3b. If PSP prefers Terminal Mode, data is sent directly to Payment Terminal)→PSP
4. Middleware sends request to PSP External Server
5. PSP External Server sends confirmation back to OPI → IFC8→ OPERA

C. Reservation Initiated from Central Reservation System (CRS) e.g. TripAdvisor, ORS, Expedia

1. Card data sent from CRS to OXI/OWS/ADS/OEDS for Token exchange
2. OXI Servers sends Card data to TPS
3. TPS Sends request to PSP host
4. PSP Host returns the TOKEN response message to TPS
5. TPS returns the TOKEN to the OXI Server.
6. OXI sends the TOKEN to OPERA CLOUD.

D. GetToken Request sent from OPERA by Entering credit car number MANUALLY into the OPERA TPS PSP for GetToken ONLY request. Bulk tokenization follows this same path. Response follows the same path in reverse direction. Follow DOTTED lines for flow *(refer to chapter 2 of OPERA s OPI User Guide CLOUD - https://docs.oracle.com/cd/F36206_01/doc.193/f36006.pdf or* V5 - https://docs.oracle.com/cd/F36206_01/doc.193/f36005.pdf)

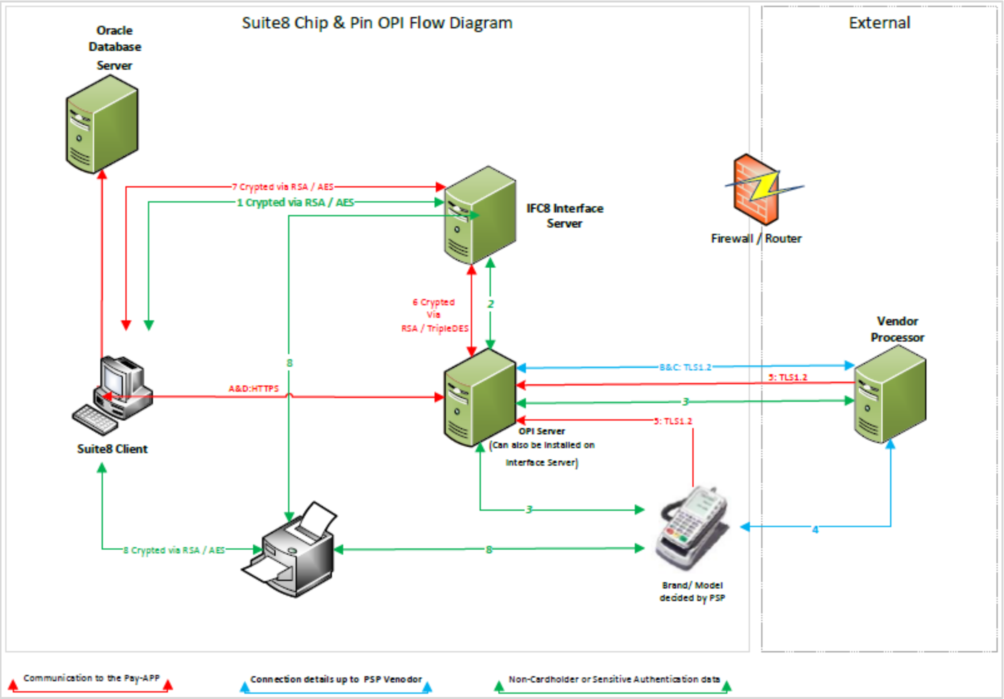
E. Token information (GetPAN if CRS not tokenized) exchange during reservation update made in opera: When a reservation is updated in OPERA PMS, a GETPAN request will be sent to PSP (PSP will have to support this feature) and the reservation will be updated in CRS using the PAN data.

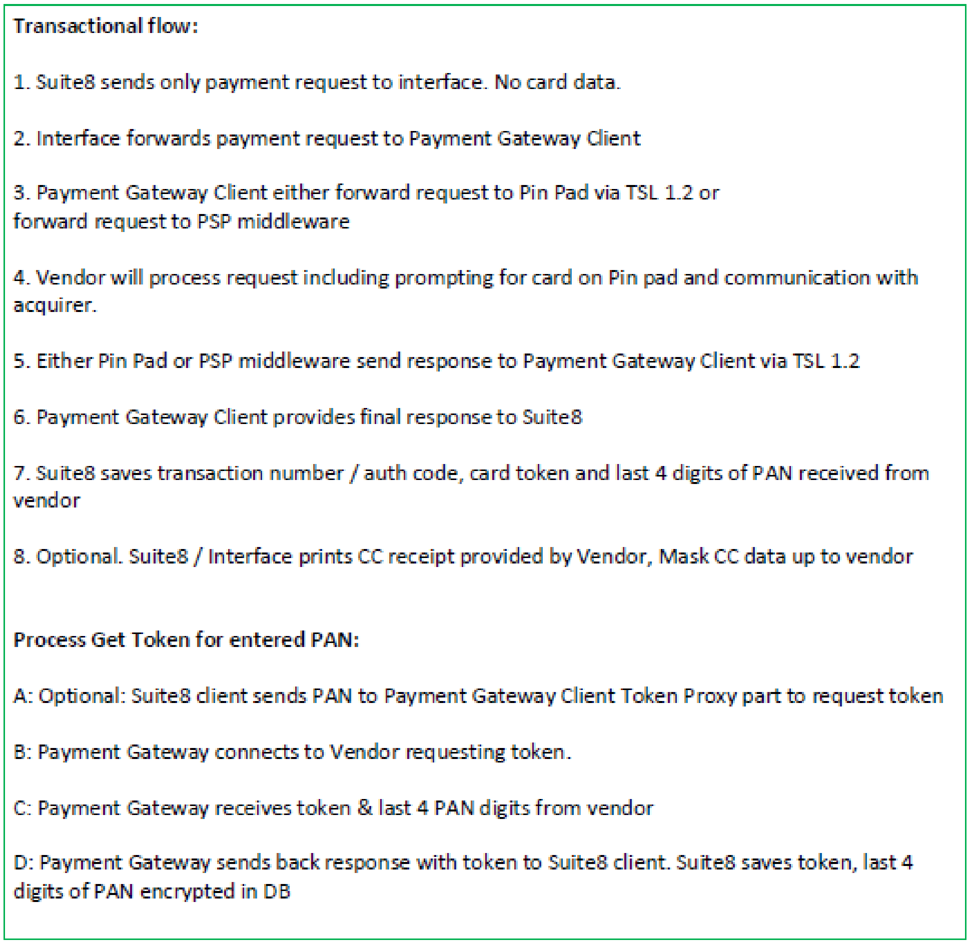
1. Opera CLOUD Sends reservation with token to OXI
2. OXI Server determines if a payment card number is needed and (OXI setting for sending token is set to "Y" or "N"). If "N" reservation with token is sent to CRS. If "Y", sends a payment card number request to the Token Proxy Server(see step 3-6)
3. TPS sends the PAN message request, with token, to PSP External Host
4. PSP Host returns the PAN response message to TPS
5. TPS returns the PAN to the OXI Server.
6. OXI sends the PAN to the External Reservation System and the reservation is updated.

**** CRS sends the UPDTATED reservation confirmation back to OXI , a new token is acquired and sent to OPERA. (refer to STEP C above)**

Suite8 Chip and Pin OPI

- Suite8 application components and database installed in hotel network, OPI will be installed in the same network.
- Install one OPI instance to process both financial transaction and token transaction.
- IFC8 will be used for communication between Suite8 and OPI.


Suite8 Chip & Pin OPI Flow Diagram
External
Oracle Database Server
7 Crypted via RSA / AES
1 Crypted via RSA / AES
IFC8 Interface Server
Firewall / Router
6 Crypted Via RSA / TripleDES
2
8
Vendor Processor
B&C: TLS1.2
5: TLS1.2
A&D:HTTPS
3
5: TLS1.2
Suite8 Client
OPI Server (Can also be Installed on Interface Server)
3
4
8 Crypted via RSA / AES
8
Brand/ Model decided by PSP
Communication to the Pay-APP
Connection details up to PSP Venodor
Non-Cardholder or Sensitive Authentication data



Transactional flow:

1. Suite8 sends only payment request to interface. No card data.

2. Interface forwards payment request to Payment Gateway Client

3. Payment Gateway Client either forward request to Pin Pad via TSL 1.2 or
forward request to PSP middleware

4. Vendor will process request including prompting for card on Pin pad and communication with acquirer.

5. Either Pin Pad or PSP middleware send response to Payment Gateway Client via TSL 1.2

6. Payment Gateway Client provides final response to Suite8

7. Suite8 saves transaction number / auth code, card token and last 4 digits of PAN received from vendor

8. Optional. Suite8 / Interface prints CC receipt provided by Vendor, Mask CC data up to vendor

Process Get Token for entered PAN:

A: Optional: Suite8 client sends PAN to Payment Gateway Client Token Proxy part to request token

B: Payment Gateway connects to Vendor requesting token.

C: Payment Gateway receives token & last 4 PAN digits from vendor

D: Payment Gateway sends back response with token to Suite8 client. Suite8 saves token, last 4 digits of PAN encrypted in DB