TL;DR?

SUMMARY ON INTEGRATION, 3 ENDPOINTS AND IPN

INTEGRATION

1) Hosted payment page or normal integration: HTTP POST/GET with all required parameters, either to default payment channel page or to specific channel page, merchant can allow or disallow the buyer to change payment method (PDF)

2) Seamless Integration: popup page or lightbox on merchant checkout flow, as simple as modifying the JavaScript snippet that could be found on GitHub

3) Mobile xdk: support variety of mobile development platforms and available on GitHub

4) Direct Server API: server-to-server request that allows merchant to handle all the UI/UX with higher flexibility and controllability (PDF)

5) Recurring API: server-to-server request that allows merchant initiated transaction (MIT) to debit buyer account anytime (PDF)

6) Offline Payment API: For in-store e-wallet payment acceptance on POS/terminal (PDF)

7) ISO Message Interface: For card acceptance terminal vendor (restricted PDF)

3 ENDPOINTS for payment response notification (For integration 1-6 only)

1) Return URL: realtime web browser or frontend direction endpoint for hosted page, seamless integration, and shopping cart module

2) Notification URL: real-time server-to-server or backend endpoint for all kind of integrations

3) Callback URL: defer update or callback endpoint on non-real time payment such as Razer Cash

IPN

1) **Frontend** IPN, applicable to return URL endpoint only: using JavaScript, you may copy from the snippet code from your merchant portal, note that **treq=0**

2) **Backend** IPN for return URL and notification URL: post back all parameters with additional parameter **treq=1** to **returnipn.php**

3) IPN for **callback** URL: just echo "**CBTOKEN:MPSTATOK**", without the quote(")

Comparison Chart

For **online** payment, merchant to decide which approaches fit the business requirements

Integration approach	**Hosted payment page**	**Seamless integration**	**Inpage checkout**	**Mobile xdk**	**Direct server API + CSE**	**Recurring API**
Browser dependency	Yes, popup and redirection	Yes, popup and redirection	Yes, iframe, popup and redirection	Smartphone built-in browser	Depends	No
UI/UX	Moderate	Better	Better	Best for mobile	Handled by merchant	Handled by merchant
Time to market (man-day)	1-7	2-14	2-14	2-14	5-60	10-90
Suitable for	Fast and easy deployment	Better UX	Better UX	In-app purchase	Better UI & UX	Subscription or billing
Mobile readiness	RWD	RWD	RWD	Native / hybrid	Backend only	Backend only
PCI-DSS compliant	PG	PG	PG	PG	PG / Merchant if handling PAN	PG / Merchant if handling PAN
Availability on github	20/22 of the cart modules	6/22 of the cart modules	Yes	12 dev-tools supported	nil	nil
Available endpoints	all 3 endpoints	all 3 endpoints	all 3 endpoints	notification & callback URL	all 3 endpoints	notification & callback URL
Fraud screening	By PG	By PG	By PG	By PG	By merchant	By merchant

For response handling, setup these endpoints(webhook) to capture the payment response

Response endpoint	**Return URL**	**Notification URL**	**Callback URL**
Browser dependency	Yes	No	No
Payment type response	Realtime payment	Realtime payment	Cash payment / defer update
Reliability	Low	High	High
Security	Low if payment verification is not enabled	High	High
IPN implementation	Frontend: easier to implement using JS snippet with treq=0 Backend: POST back all values + treq=1 to returnipn.php, more reliable	Backend: POST back all values + treq=1 to returnipn.php	Backend: echo “CBTOKEN: MPSTATOK”