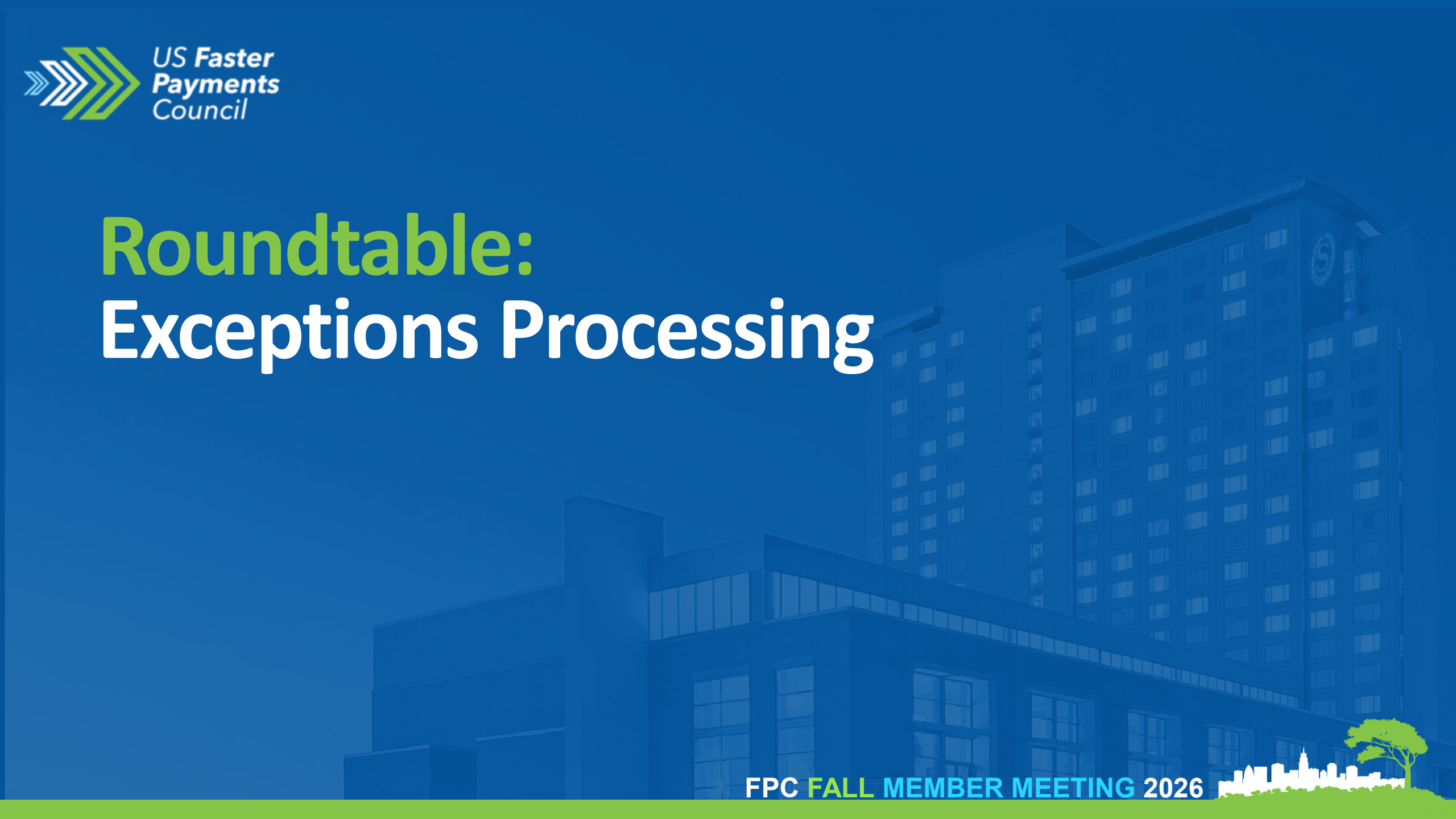
US Faster Payments Council

FPC FALL MEMBER MEETING 2026



US Faster Payments Council

Roundtable:
Exceptions Processing



FPC FALL MEMBER MEETING 2026

US Faster Payments Council

Roundtable: Exceptions Processing



Tracy Merritt
Senior Vice President
ePayAdvisors



Stephanie Miracle
Director, FedNow Service Product Management, Federal Reserve Bank of Boston



Jamie Florio
Director of Strategic Accounts
Fincom



Kevin Clark
SVP, Strategic Operations Officer, Corporate One FCU

Exception Workflow



Initiating FI

FedNow 24/7 INSTANT PAYMENTS

R | T | P



Receiving FI





OFAC

AGNT	99##
AM02	DS24
AM13	FF02
DT04	RC01

AC03	AG01	AM01
AC04	AG03	
AC06		
AC14		





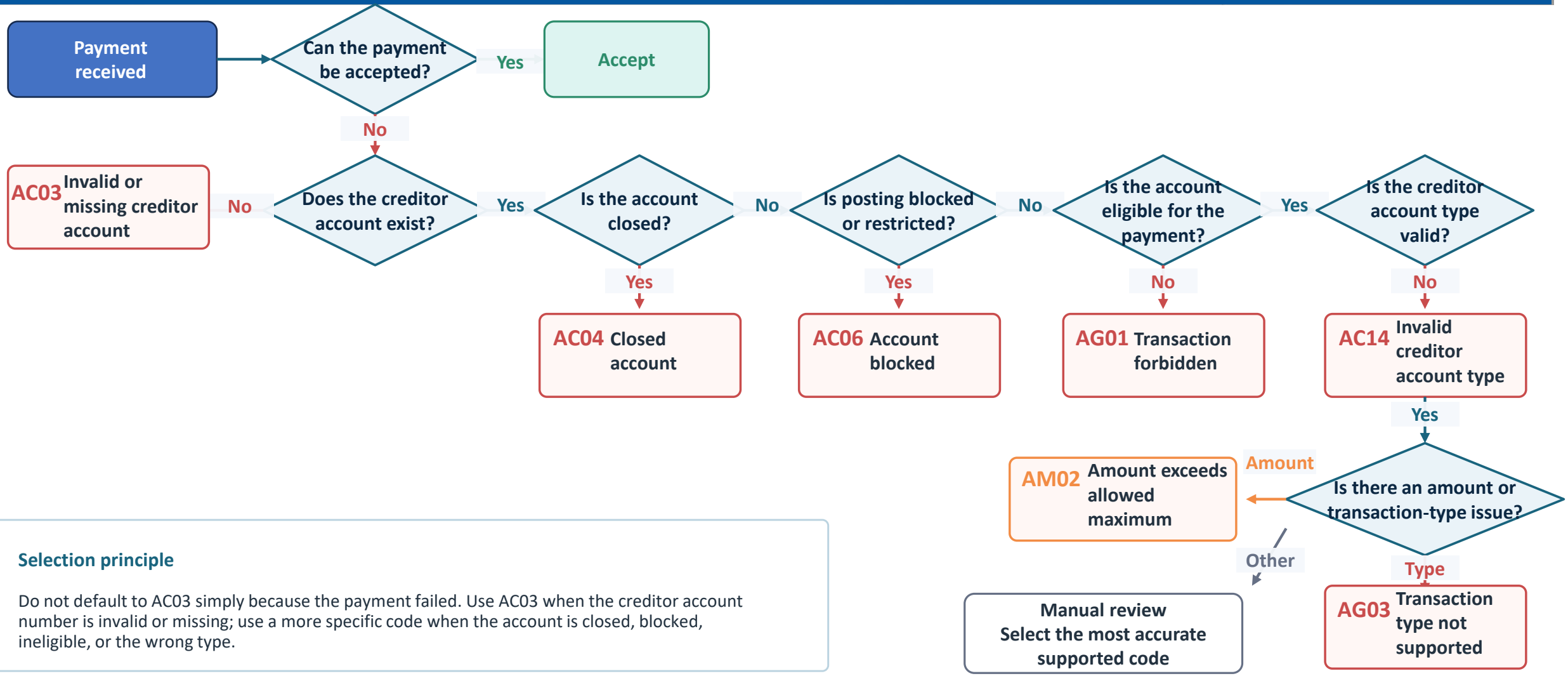
OFAC

US Faster Payments Council



Detailed transaction rejection flow

Use the first condition that accurately describes why the credit transfer cannot post.



Selection principle

Do not default to AC03 simply because the payment failed. Use AC03 when the creditor account number is invalid or missing; use a more specific code when the account is closed, blocked, ineligible, or the wrong type.

Reject code ISO quick guide

A companion reference for the decision flow

AC03 — **Invalid or missing creditor account**
Use when the receiving account number does not exist or is missing.

AC04 — **Closed account**
Use when the receiving account existed but has been closed.

AC06 — **Blocked account**
Use when posting is prohibited because the account is blocked or restricted.

AG01 — **Transaction forbidden**
Use when this payment is not permitted for the account.

AC14 — **Invalid creditor account type**
Use when the receiving account type is not eligible.

AG03 — **Transaction type not supported**
Use when the payment or message type is unsupported or unauthorized.

AM02 — **Amount exceeds maximum**
Use when the transaction amount is greater than the allowed maximum.

REVIEW — **No clear match**
Escalate rather than forcing a generic code; confirm the network and processor implementation guide.

US Faster Payments Council



Take Aways

- **Learn from the exceptions you already handle in other rails**
- **Find the manual workarounds that hide existing weaknesses**
- **Push technology providers for better support for “immediate”**
- **Don’t let your technology limitations limit your approach**

