curbito
CURB INFO TO OWNERS

TECHNICAL DISCOVERY REQUEST

Quiet meter-session orchestration

A synthetic-first proposal for Minneapolis Parking Services and Flowbird: recognize existing sessions, suppress duplicate reminders, preserve the official payment channel, and surface only the curb conflicts another meter app cannot see.

FROM	Robert Sims, Founder, Curbito
EMAIL	hello@curbito.com
PHONE	612-481-8223
PILOT MARKET	Minneapolis, Minnesota

curbito.com · hello@curbito.com

Request

Curbito requests a technical discovery meeting with Minneapolis Parking Services and Flowbird to evaluate a limited, synthetic-first integration with the MPLS Parking system. Curbito does not propose replacing the official app, processing parking revenue independently, or sending duplicate meter-expiration reminders.

The proposed role is a quiet orchestration layer:

1. After a likely parking stop, resolve the controlling curb rule and exact paid space.
2. Ask the authorized provider whether the vehicle already has a valid session for that space.
3. Remain silent when the official provider session covers the stay.
4. Leave routine expiration reminders with the official provider.
5. If payment is missing, open a provider-owned checkout or approved deep link with the exact space prefilled, subject to driver confirmation.
6. Alert through Curbito only when a separate verified curb restriction conflicts with an otherwise valid payment session.

This distinction matters because a paid meter session does not override Snow Emergency rules, street sweeping, temporary no-parking signs, loading restrictions, construction restrictions, or other controlling instructions.

Requested discovery items

We are asking whether the existing Minneapolis/Flowbird configuration can support any of the following through documented, contractually authorized interfaces:

- Meter-zone and numbered-space inventory with geometry, hours, rates, maximum stay, accessible rules, event rates, and current effective timestamps.
- A least-data active-session lookup using a provider-issued account/vehicle token, zone or space identifier, and session expiration—without bulk plate access.
- A provider-owned checkout URL, app deep link, SDK, or API quote that can prefill the exact zone/space and vehicle while keeping the City or designated provider as merchant of record.
- Apple Pay availability inside that approved checkout flow, rather than a separate Curbito payment vault.
- Signed callbacks for session started, extended, expired, cancelled, and payment failed.
- Sandbox credentials and synthetic zones, vehicles, and sessions for conformance testing.
- Provider notification metadata sufficient to suppress duplicate Curbito expiration alerts.
- Error codes for ambiguous space, maximum stay, closed zone, rate change, invalid vehicle, and provider outage.
- Service-level, rate-limit, support, audit, and incident-notification expectations for a small pilot.

Proposed minimum data contract

Curbito to provider

- Opaque Curbito installation or user token.

- Provider-issued vehicle/account token where authorized.
- Exact provider zone and numbered space when required.
- Requested parking duration.
- Idempotency key and request timestamp.
- User confirmation that immediately precedes a charge or checkout handoff.

Provider to Curbito

- Provider, zone, and space identifiers.
- Session state and expiration.
- Signed quote or provider checkout location.
- Maximum allowed duration and price supplied by the provider.
- Opaque provider session/receipt identifier.
- Signed callback event ID, timestamp, lifecycle state, and retry metadata.

Excluded by default

- Bulk plate scans or enforcement search access.
- Full payment-card data.
- Provider account passwords.
- Enforcement-officer identity or internal notes.
- Unrelated parking or travel history.
- Automatic recurring or silent charges.

Accuracy and user safeguards

- GPS alone will not be treated as proof of a numbered meter space. If the provider cannot supply a deterministic space match, Curbito will request a sign/QR/NFC scan or a one-time space confirmation.
- No payment prompt appears until the current curb remains eligible for parking.
- No checkout proceeds without an exact provider space and explicit user confirmation.
- A provider acknowledgement is required before Curbito records a session as paid.
- Provider-owned expiration reminders suppress Curbito duplicates.
- Stale, nearby-only, raster-derived, or incomplete rules cannot produce an all-clear or initiate payment.
- Posted signs, official instructions, and provider records remain authoritative.

Synthetic pilot proposal

1. **Contract review:** confirm use case, data classification, merchant-of-record boundaries, and approved interface.

2. **Sandbox:** test ten synthetic zones and vehicles covering active session, missing payment, ambiguous space, maximum stay, future restriction, cancelled zone, and provider outage.
3. **Staff demonstration:** run the Curbito partner portal and native app against synthetic signed callbacks.
4. **Limited field pilot:** opt-in founder/City test accounts in a small geographic area, with no automatic charges and daily audit review.
5. **Evaluation:** measure incorrect prompts, duplicate alerts, successful handoffs, provider errors, battery impact, user action time, and support volume before any expansion.

Standards alignment

Curbito's curb-decision boundary is designed to consume Open Mobility Foundation Curb Data Specification structures and keep provider-specific payment logic behind a separate adapter. This allows Minneapolis curb regulations and Flowbird session/payment records to remain independently authoritative while producing one user action.

Questions for Minneapolis and Flowbird

1. Which organization owns approval for each requested interface: Minneapolis Parking Services, City IT/data privacy, Flowbird, or another contracted operator?
2. Does the current Flowbird configuration use pay-by-space, pay-by-plate, or both for mobile transactions, and what exact identifier does enforcement query?
3. Is there an existing partner API, deep-link, SDK, or authorized aggregator program for third-party user interfaces?
4. Can an active session be checked without transmitting or returning a raw plate?
5. Can the official checkout accept a preselected space, vehicle, duration, and Apple Pay while preserving the existing merchant-of-record arrangement?
6. Which provider reminder states are available so Curbito can avoid duplicate expiration notifications?
7. Are temporary closures, suspended spaces, Snow Emergency overrides, and event-rate changes exposed through the payment platform, the City's curb-data work, or a separate source?
8. What security review, insurance, accessibility, procurement, records, and pilot agreement requirements should Curbito prepare?

Demonstration

- Public product site: https://curbito.com
- Synthetic partner workflow: https://curbito.com/partners/portal
- Contact: hello@curbito.com

Curbito welcomes a narrow discovery call even if only a subset of the requested capabilities is currently available. The immediate goal is to learn the correct system boundaries and demonstrate a safe synthetic workflow before requesting any production data.

Public basis for this request

- The City of Minneapolis parking-meter instructions identify numbered spaces, direct drivers to the MPLS Parking app, and state that Snow Emergency and street-sweeping rules override meter use.
- The current MPLS Parking FAQ states that app accounts and alert settings are managed by Flowbird.
- Flowbird's e-ticket platform description says third-party mobile-payment channels can be integrated when the necessary interfaces are implemented.
- The Open Mobility Foundation CDS repository identifies CDS 1.1 as the current release and describes the Curbs API as the standard for publishing curb locations and parking regulations.
- The Open Mobility Foundation lists Minneapolis in the 2026 Smart Curb Collaborative, whose work includes expanded CDS implementation and adoption.