LNP/BP Standards Association

Universal LNP/BP invoices

Draft standard proposal (LNPBP-38)

LNP/BP Standards Association
Proposal by **Dr Maxim Orlovsky**

Sponsored by **Pandora Core AG**

State of payments in Bitcoin ecosystem

- Bitcoin **addresses** (2 standards so far + update to bech32)
  - promote bad practices (pubkey reuse)
- Bitcoin URLs with amounts (**BIP-72**)
  - rely on addresses
  - not copyable with a single click
- Lightning invoices (**BOLT-11**)
- **LNURL** initiative
  - interactive protocol
  - relies on Internet1 standards :(
- New Lightning invoicing protocol by Rusty Russel (**BOLT-14**?)
  - still lightning-only
- **RGB** invoices (**LNPBP-37**) (URL based, after Alekos Filini)
  - LN-incompatible
  - not copyable with a single click
  - limited payment options/scenarious

Problems with invoices today

- Payment-channel specific
- No support for multiple asset types
- Confidentiality leaking
- No extensibility
  (both in terms of protocol upgrades or custom vendor extensions)
- Infexible encodings
- Very limited functionality
- Low protection

Universal invoice structure

- One or more **beneficiaries**, ordered by payee preference
- Used **network** magic byte (not genesis hash but P2P network id)
- List of **optional fields** structured as TLVs in LN
  - if the field is absent, no space occupied
  - each field has a type id
  - fields with even type id must be understand by the payer
  - fields with odd type id may be ignored if now known to the payer
  - types may be standard (defined as LNPBPs) or vendor-specific

Universal invoice: “beneficiaries”

- “Legacy” **bitcoin addresses** (all types, including future)
- RGB **blinded UTXOs**
  (hash of txid:output_no + 64-bit salt, used as payment secret and kept by payee)
- Descriptors with custom **miniscript**: can be used for automatic address derivation by the payer
- **PSBT**-based: for payment aggregation or for simplifications of payments from multisig addresses (with hardware devices & multiple participants)
- **Lightning**: specifies receiver node id + address (IPv4, v6, Onion v2, v3)

Beneficiary/“address” types comparison

Feature	Classical addresses	Blinded UTXO	Miniscript descriptors	PSBTs	Lightning “addresses”
Repeated / “anyone can pay” payments	possible, but privacy leaking	with expiration date only	very good	not possible	very good
Confidentiality	lowest	very good	fine (non-hardened branch of pubkeys exposed)	single UTXO exposed	good
Transaction batching (non-RGB)	not possible	not possible	very good	very gooo	not needed
Multisigs, HSMs & payjoins	not possible	not possible	very limited	very gooo	not applicable
Can be used by payee w/o UTXOs	not applicable	not possible	yes	yes	not applicable
Size footprint	small	small	moderate	significant	significant

“Lightning address” concept

- Fields from BOLT-11 specific to payee
  - Node id (public key of the node)
  - Node features
  - Hash lock
  - CLTV expiry
  - path hints (list, may be empty)

- Other fields become part of fields shared across different payment options (beneficiaries)

Optional standard fields (current proposal)

- **asset_id** – 256-bit hash
  - equivalent for LN ChainCode or RGB contract_id
  - allows to use protocol with different blockchains (liquid) &
    assets systems (confidential assets)
  - required for RGB, otherwise defaults to bitcoin
- **price** – a price per item
  - instead of amount
  - based on Rusty Russel ideas for new LN invoice protocol
  - the payment can be a multiple of the price (see quantity below)
  - zero price signifies that the amount have to be determined by the payer
    - charity
    - invitation to open a channel
- **quantity** – limits how many units may be bought. Consists of fields:
  - minimum (optional, defaults to 0)
  - maximum (optional)
  - recommended (one if not specified)

Quantity is very useful for micro/nanopayments (per second of the video, per message, per km of autobahn)

- **fiat_requirement** – specifies asset price bound after which merchant MAY change terms & conditions
  - ISO4217 code for fiat equivalent
  - round and fractional part of the amount in that currency
  - URL of the price provider (protocol defined by the schema; for HTTP defaults to a text response with a single price tag

Optional standard fields (current proposal)

- **purpose** – a string describing the invoice
  - 639 chars max, as per BOLT-11
- **details** – external details
  - commitment hash (double SHA256), BOLT-11 `h` tag equivalent
  - URL (URL schema defines protocol; https(s) defaults to text/plain response);
    - may be empty for BOLT-11 compatibility
    - text response may be GPG/PGP signed with the same key as used in the invoice signature
- **merchant** – a string identifying merchant for UI purposes
  - max 32 chars
- **expiration** – UNIX timestamp (signed 64-bit integer) defining GMT time when the invoice will not be valid
- **signature** – Schnorr public key + signature over merkle tree root of invoice fields

Key properties

- By default, non-interactive, but with multiple payment options and scenarios supported
- Can be constructed from LN BOLT-11 invoices or transformed into BOLT-11 invoice
- Optionally signed with merkle tree of TLVs, such that signature can be revealed without revealing full of the invoice
- For QR may be reduced in size by
  - skipping odd TLV fields
  - leaving only single payment option (beneficiary)
  - removing path hints from lightning beneficiary
  - skipping signature

Universal invoice encoding

- Binary structure
- When transferred as a LN message encoded according to LN message rules (by design it is compatible with them)
- When used with RGB, encoded using strict encoding rules
- For humans, encoded as bech32 string with `i` prefix
- May be QR-encoded as a uppercase bech32 string with ‘lnpbp:’ URL schema

Roadmap

- Draft a standard text
- Prepare draft implementation in **LNP/BP Core Lib v0.3**
- Have a community discussion
- Release language-specific libraries based on LNP/BP Core (**libinvoice** C Lib, lnpbp-invoice NPM, InvoiceKit, …)
- Ship as a part of **RGB Node v0.3** and **LNP Node v0.2**
- Include in **Bitcoin Pro** (advanced invoice editing) & **MyCitadel** wallet
- Look for others devs & industry to adopt it

Opened questions

- Best way to support channel opening proposal
- Best practices for automatic address derivation in miniscript/PSBTs
- Bech32 prefix (HRP)
- QR encoding URL prefix
- Confidentail assets compatibility
- Terminology (fiat -> currency, many be other better terms)
- Payment splitting between beneficiaries
- Payments in multiple assets - do we need them?

Materials

- Discussion: https://github.com/LNP-BP/LNPBPs/discussions/82
- Initial implementation: https://github.com/LNP-BP/rust-lnpbp/pull/165