Goal:

- This internal document should form the foundation of a UMIP that proposes a standard format for the ancillary data of any price request. By conforming to this format, the price requester makes it easier for UMA voting clients to more easily translate this ancillary data into readable form for voters.

Interface Requirements:

- Must be able to display arbitrary # of price request parameters that can be used to instruct voters how to resolve a given price request for an identifier and timestamp. For example, this should enable UMA voters to resolve the SPORT_X identifier for different competitions across different sporting events
- Must fit into 8192 bytes, as enforced by the Voting smart contract.
- Must be easily appendable at the smart contract level so that different smart contracts can append data to the ancillary data. For example, if the initial ancillary data is "competition:World Cup,date:07-09-2024,match:FRA-ESP", then the financial contract might want to append "requester:<CONTRACT-ADDRESS>". This stamping of the requester address allows the DVM to combat parasitic usage. This could be done via an internal method like *_getAncillaryData(): { return abi.encodePacked(ancillaryData,",Requester:",address(this)) }*

Proposal for parsing ancillary data from bytes to utf8

- if any step fails, then voter dApps will default to displaying raw byte data:
- Convert bytes to UTF8
    - Converting different data types to UTF8 in Solidity can be complex. This is a Solidity library that provides methods to convert `uint` and `address` types to utf8-decodable strings, for example.
- Delimit UTF8 string by "," (commas) into a list of key-value pairs:
- Each key-value pair can be further delimited by ":", colons
    - The author of the UMIP for the associated identifier is free to define how the values in these key-value pairs should be formatted in UTF8. For example, one identifier might want dates in MM/DD/YYYY format, and another might want them in unix time.

Exceptions to the special delimiter characters:

- **Skip delimiters inside unclosed double quotes:** Some ancillary data might include special characters that collide with the "," and ":" characters that we use to delimit the key-value dictionary. Therefore, its important that we have a mechanism for escaping such characters in the ancillary data (which could easily appear in URL's for example).

- Example ancillary data:
  - title:"ethusd,sourceURL:https://www.binance.com/en/trade/ETH_USDT"
    - *The ":" in the sourceURL value is ignored.*

- **Skip delimiters inside unclosed square brackets ([]) and curly braces ({}):** Developers might want certain value strings to be parsed directly as JSON. For example, voters could use JSON values to construct a pricefeed with which to compute prices. Therefore, any commas or colons inside (1) unclosed quotes or (2) unclosed brackets should be ignored.
  - Example ancillary data:
    - title:"someDictionary,json:{k1:v1,k2:v2}"
      - *The ":" and "," in the json value are ignored*
    - title:"priceFeedConfiguration",priceFeed:{"type": "expression","expression": "ETHBTC_FV = ETH\\/BTC * PERP_FRM; round(max(-0.00001, min(0.00001, (ETHBTC_FV - ETHBTC_PERP) / ETHBTC_FV / 86400)), 9)","lookback": 7200,"minTimeBetweenUpdates": 60,"twapLength": 3600,"customFeeds": {"ETHBTC_PERP": {"type": "uniswap","invertPrice": true,"uniswapAddress": "0x899a45ee5a03d8cc57447157a17ce4ea4745b199"},"PERP_FRM": {   "type": "frm","perpetualAddress": "0x32f0405834c4b50be53199628c45603cea3a28aa"}}}"
      - *This ancillary data could specify which price feed configuration a bot/voter should use to compute an identifier price*
- Key-value pairs, where keys are only different by trailing incrementing natural numbers can be combined into arrays
  - Example ancillary data:
    - Perpetual-token-address:0x123,requester:0xabc
      - This could be for a funding rate identifier like ETHBTC_FR
    - competition:World Cup,date:07-09-2024,match:FRA-ESP,requester:0xabc
      - This could be for a Sport-x catch-all identifier
    - Strike:2500,expiry:1626732098
      - This could be for a ETHUSD identifier for a call option contract
    - Id0:1337,id1:2448
      - This is an array of ids [1337,2488]