what good is partial understanding?

let's say you ask me
what time it is right now

today I say
"11 04 27"

today I say
"hh mm ss"

today I say
"0..24 0..60 0..60"

I decide to "improve" this (unilaterally)

now I say
"2024 10 17 11 04 27 014"

now I say
"yyyy mm dd hh mm ss sub"

if you were the old program
you would be very confused

our formats are NOT self describing:
iykyk

fine let's make it
self describing

11 04 27

11h 04m 27s

11h 04m 27s

the messages are bigger

but now we understand

then what if I send
2024y 10M 17d
11h 04m 27s 014i

then what if I send

2024y 10M 17d

11h 04m 27s 014i

neat

then what if I send
27s 11h 04m

then what if I send

27s 11h 04m

I CAN understand this

I CAN'T understand this
in one pass

27s 11h 04m

we've gone from decoding
to querying

but what if I send

11h 04m

now we have to admit
queries can fail

or what if I send

"eleven"h "four"m "twelve"s

now we have to admit
types are part of our queries

messages can be
well formed
parseable
queryable
and yet still be insufficient

self describing formats are basically all key:value stores

JSON, TOML, YAML, ProtoBuf
CBOR, ASN, etc.

for better or worse, they allow failable queries and bonus data

did this help us?

maybe.

but you could just...

here be dragons

and if we have the discipline
to keep messages
"semver compatible"

couldn't we just admit
we have more than one
kind of message?

self describing formats
have benefits

but perhaps the costs
are a bit more nuanced