

Invalidating Offers

And making the API simpler?

Issue #366

Problem: When is an offer valid?

- Problem code:
 - `createOffer().then(a => globalA=a; return createOffer()).then(b => setLocalDescription(globalA))`
 - Not clear why anyone would ever do that, but it's not explicitly disallowed at present.
 - Also not defined to be sure to work, because of the “different stable states” issue
- A problem because things may have changed between the two `createOffer` calls.

Other options considered

Or - how did we get here?

Solution A: Invalidate old offers

- CreateOffer() makes all older offers invalid
 - And similar for CreateAnswer()
- No changes to existing code

Solution B: Commit to the offer sent to setLocalDescription

The “problem code” would continue to work as it does today, but subsequent calls to `setLocalDescription(b)` would not work.

Not clear that this is a good idea (see “problem”).

Solution C: Remove SetLocalDescription

The only API call would be “offer”, which returns a description, and also commits it as the local description

Cons:

- Not compatible with existing code
- Can't inspect offer before committing it locally

Solution D: Change setLocalDescription argument

SetLocalDescription always commits to most recently generated offer or answer

Could keep argument in WebIDL for legacy purposes (and checking that people weren't trying to change it)

Solution Step 1

Add support to browsers for this:

```
setLocalDescription("offer") // Or "answer" or "pranswer"
```

It applies the last created offer or answer (or does a rollback).

Solution Step 2

Tell app developers to replace this:

```
setLocalDescription(offer)
```

With this:

```
setLocalDescription("offer")
```

And warn them that `setLocalDescription(offer)` will blow up if they pass in the wrong one

Solution Step 3

Once enough time has passed for apps to transition,

Blow up if `setLocalDescription(offer)` does not have the same offer that would have been applied by `setLocalDescription("offer")`

Pros

- Doesn't break any app code that is guaranteed to work today
 - (or even the dodgy ones, given enough time for them to update)
- App code is actually more simple
- The spec is more simple

Cons

- It's a bit implicit. `createAndSetLocalOffer` would be more explicit, but that's probably going too far at this point.

Further down that rabbit hole...

- `setLocalDescription("offer")` could create an offer if you haven't created one yet.
`"pc.setLocalDescription("offer").then(=> signalOffer(pc.pendingLocalDescription))"`
- `setLocalDescription("answer")` could create an answer if you haven't created one yet.
`"pc.setRemoteDescription(offer).then(=> pc.setLocalDescription("answer").then(=> signalOffer(pc.currentLocalDescription))"`
- `setLocalDescription()` could deduce "offer"/"answer" from the signaling state (stable => "offer"; have-remote-offer => "answer")

Subgroup recommendation:

Option D