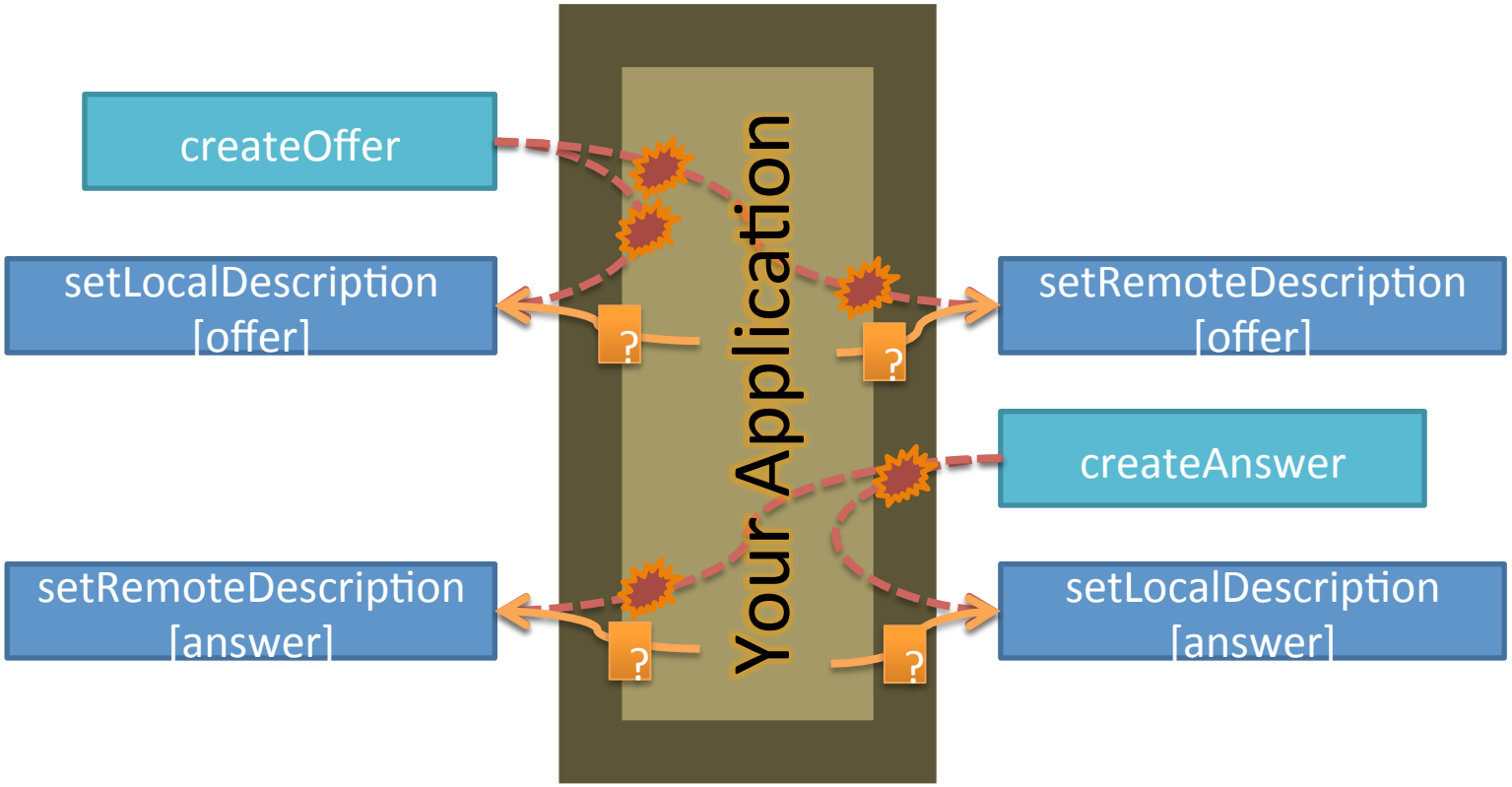
Waiter, I appear to have SDP in my API!

Photo by Name: Anders Dahnielson
See : http://www.flickr.com/photos/dahnielson/379519464/

Problem Definition



1) What SDP does createOffer & createAnswer produce
2) What ways are OK to change this SDP before passing to browser
3) What errors does the consumer produce

Goals

- A clear definition of SDP for
  - Interoperability
  - Use cases for modifying SDP
  - Application certainty on modification to SDP
- Error handling rules
  - Reject or ignore errors on input
  - Probably both, depending on class of error
    - Therefore, need rules describing which
  - Error reporting we can handle

Alter SDP at own risk?

- This option doesn't seem realistic

+ Lots of latitude for innovation
- SDP could effectively become immutable
  - We rejected that idea when we chose JSEP
- No real hope for interoperability
  - setRemoteDescription needs a stricter definition than this approach would provide

Browser Modifications

- The following are true
- ```
  pc.setLocalDescription(desc);
  assert(pc.localDescription.sdp === desc.sdp + new
  candiates );
  ```
- and:

  ```
  pc.setRemoteDescription(desc);
  assert(pc.remoteDescription.sdp === desc.sdp + new
  candiates );
  ```
  ```

What about Constraints?

- Constraints can enable common use cases
- Constraints do not resolve the issue of what can be changed in SDP

- Proposed approach:
  - define “SDP”
  - add constraints for the identified use cases where it makes sense

SDP Definition

- Need a definition of what SDP needs to be supported for the output of createOffer/createAnswer and the input to setLocalDescription/ setRemoteDescription

- **For each possible SDP line (or parameter), we need rules for…**
- Presence, SDP:
  - MUST include
  - MUST NOT include
  - MAY include (or SHOULD, whatever)
- Processing rules; for all “MAY include” rules, if not supported or possible then:
  - MUST reject
  - MAY ignore (SHOULD, blah blah)
  - MUST ignore
  - Could be different rules for offers and answers
- This type of information is being added to JSEP I-D

What SDP can be changed from create Offer/Answer to set Local/Remote?

- Use cases:
  - Remove a codec
  - Change bandwidth limit (note limit not target)
  - Change resolution / frame rate
  - Change SDES
- Lets have a a better way than SDP mangling
  - Constraints
  - Specific API methods

“Negotiation” failure

- Offer / answer defines what happens in negotiation
- For example:
- Offer G.711 + VP8
- Receiver does not do video
- Answer with G.711

When can two different video flows use the same m-line?

- From the HTA proposal
- IFF all the following is true
  - All codec parameters are the same
  - The “content-label” is the same
  - They are in same MediaSteam

How does createOffer know to offer receiveOnly flow?

- Consider a browser that did not have a camera but wanted to receive 2 streams of video corresponding to the content label “main” and “slides”
- HTA how does this work ?

How does createOffer decide to offer a data channel

- Work by following: If one has been added to the PeerConnection?
- Do we want to add OfferToExchangeData constraint?

DTMF

- How does offer decide to offer DTMF and on which audio track?

How long is SDP from create Offer/Answer valid ?

- At least for duration of the callback function

Rollback Proposal

- When system does a setLocal with an offer, it must be willing to keep receiving media defined by previous offer/answer until one of following happens:
  - Rollback: setLocal with value of previous setLocal
  - New offer / answer pair: setRemote called with answer that matches the new offer. (note a pranswer does not do this)
- Other proposal?

Error Handling

- Major class of failures / warnings
  - I can't parse this SDP
    - Example: pass JSON structure to SDP
  - This SDP is not actionable
    - Example: answer does not match offer
  - Unsupported SDP extension
    - This is not an error but might be interesting to report
- Two classes of response identified:
  - Reject operation outright for error with string that provides hint of where things went wrong
  - Generate a report of ignored items

Error Feedback (defer to general time for this)

- For reject, feedback mode
  - Inline (throw new Error())?
  - Error callback?
  - Best if the API is consistent in choice of feedback mode, e.g.:
    - Purely syntactic errors are reported with exception
    - Errors arising from application of SDP are all through the callback
  - Do we want to have to deal with several ways to get errors?