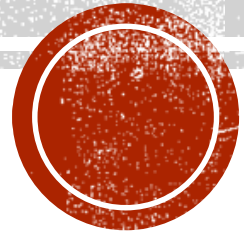
SDES

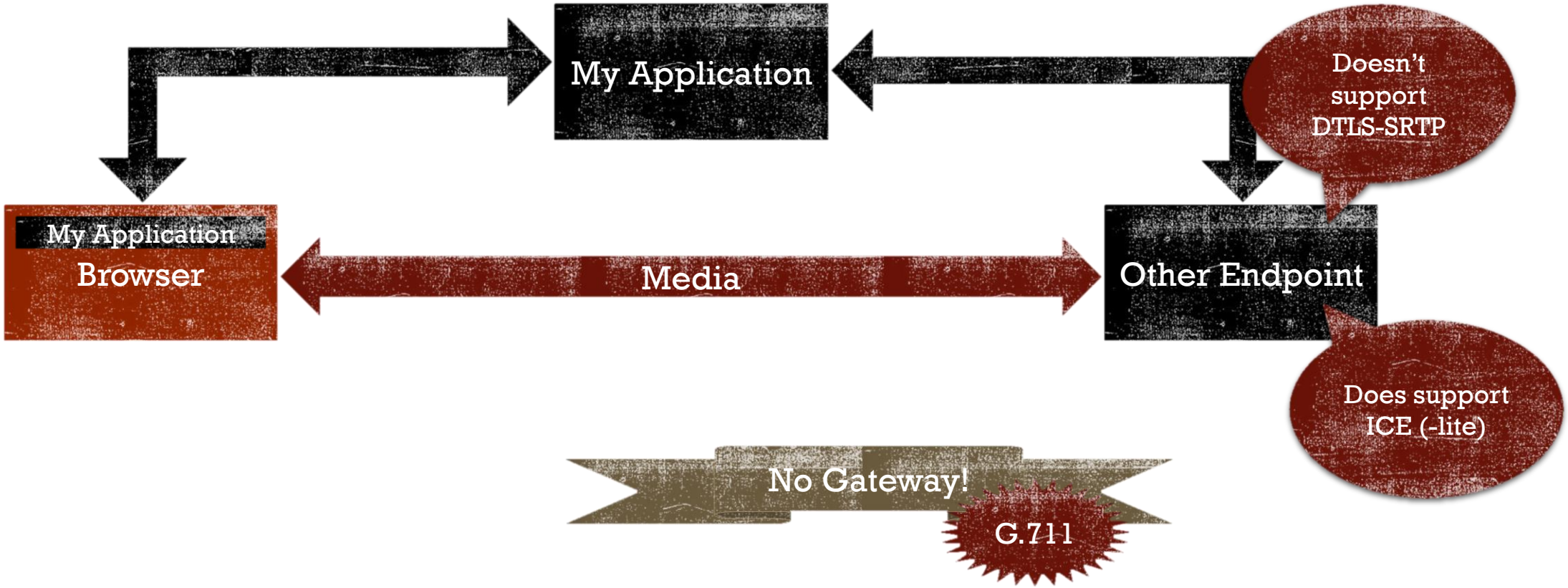
IETF 87, RTCWEB WG



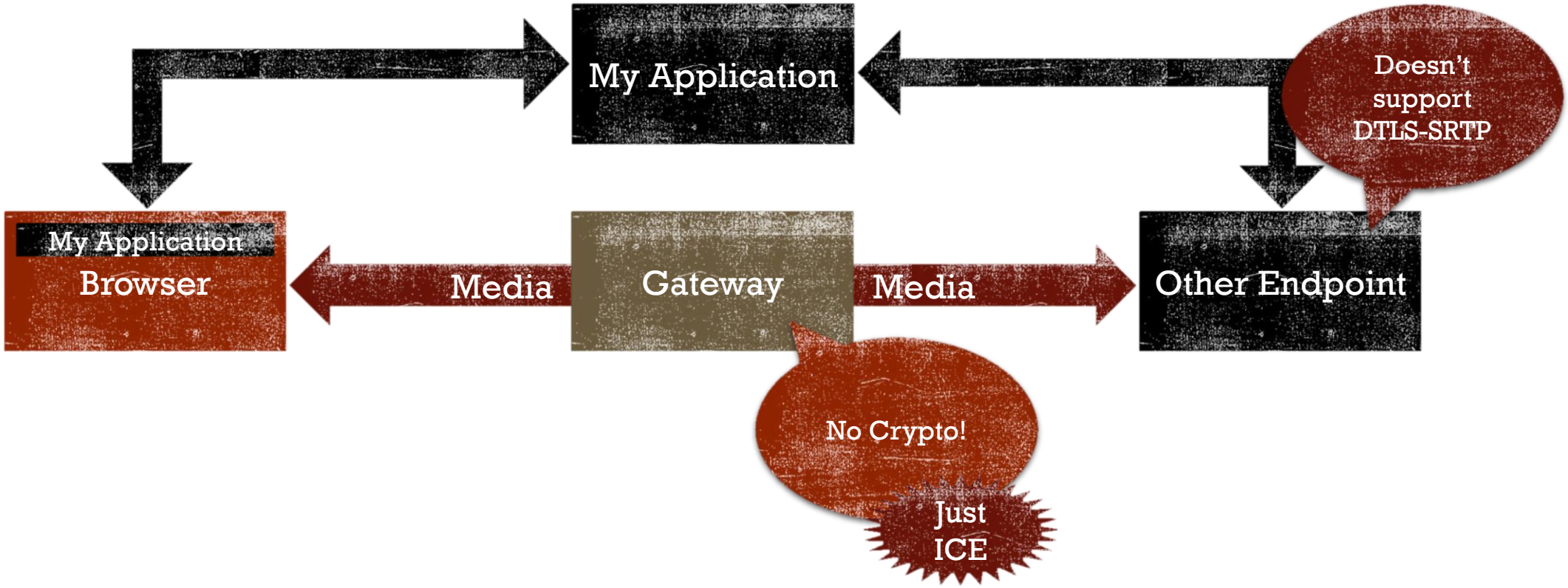
WHY I WANT TO USE SDES

- Mostly, I don't
- …but it does make my application work better, with less code

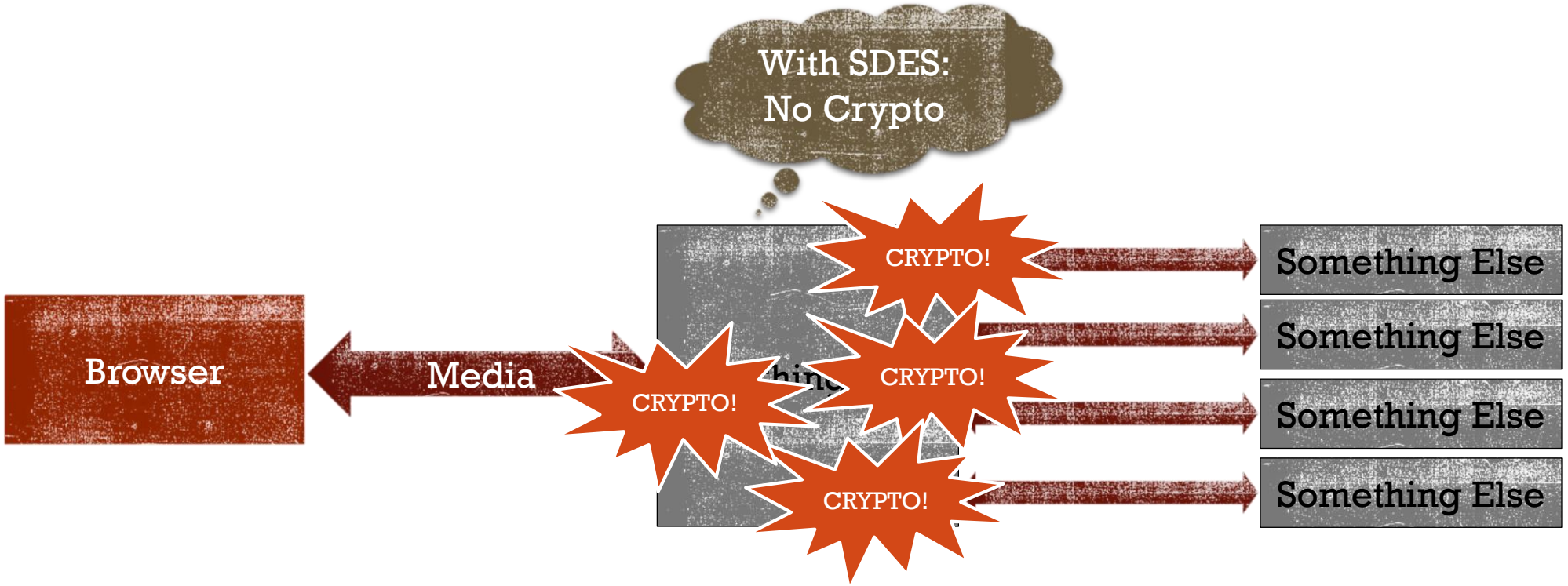
WHY SDES IS AWESOME AND USEFUL (1)



WHY SDES IS AWESOME AND USEFUL (2)



WHY SDES IS AWESOME AND USEFUL (3)



WHAT PROBLEMS THIS SOLVES

- Early media clipping
  - SDES can allow for no extra round trips*
- No crypto at gateways means media and keys aren't in same place
  - Some small advantage with respect to public perception (c.f., PRISM)
  - …maybe
- Delays the inevitable heat death of the universe
- Produces fewer CO_2 emissions

* RFC 4568 got the a=crypto parameter back to front, so you need haxx for this

WHY WE USE DTLS-SRTP

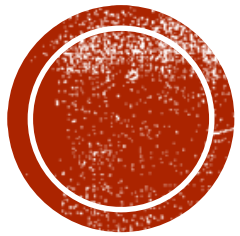
- DTLS-SRTP makes everything better
  - It's the MSG of media security protocols
- Plus, it makes it possible to authenticate the entity you are talking to
  - You can know where your voice/audio/smells are going
  - …and where that thing you are seeing/hearing/touching is coming from

WHAT DOES IT TAKE TO AUTHENTICATE?

- For SIP, DTLS-SRTP and RFC 4474 are a raging success
  - Authentication is a doddle, see STIR BOF
- In WebRTC, this relies on the identity provider model
  - Each peer needs to acquire media using noaccess or peerIdentity constraints
  - Each peer needs to acquire an identity assertion from their IdP
  - Signaling is needed to exchange identity assertions
  - Certificate fingerprint used in DTLS handshake is bound to identity using assertion
  - Provide good user feedback (around getUserMedia, remote identity, media origin)

ALL OR (CLOSE TO) NOTHING

- Miss a step and you lose
- The site can see, modify, synthesize, destroy*, etc… your media
- All you have from your DTLS-SRTP is then:
  - An extra RTT or two
  - Some identity information, which might allow **auditing**



FIN

BACKUP: THE DTLS-SRTP+EKT SLIDE