

Best Practices For Executing High Profile Live Streaming Events

Eric Zawolkow, Media Systems Engineer - Facebook

James Broberg, Founder, CEO - StreamShark.io (james.broberg@streamshark.io)

DUE TO THE **GRAPHIC NATURE** OF THE FOOTAGE,
THE FOLLOWING **TRAILER** HAS BEEN
SPECIALLY MODIFIED FOR ALL HUMANS.

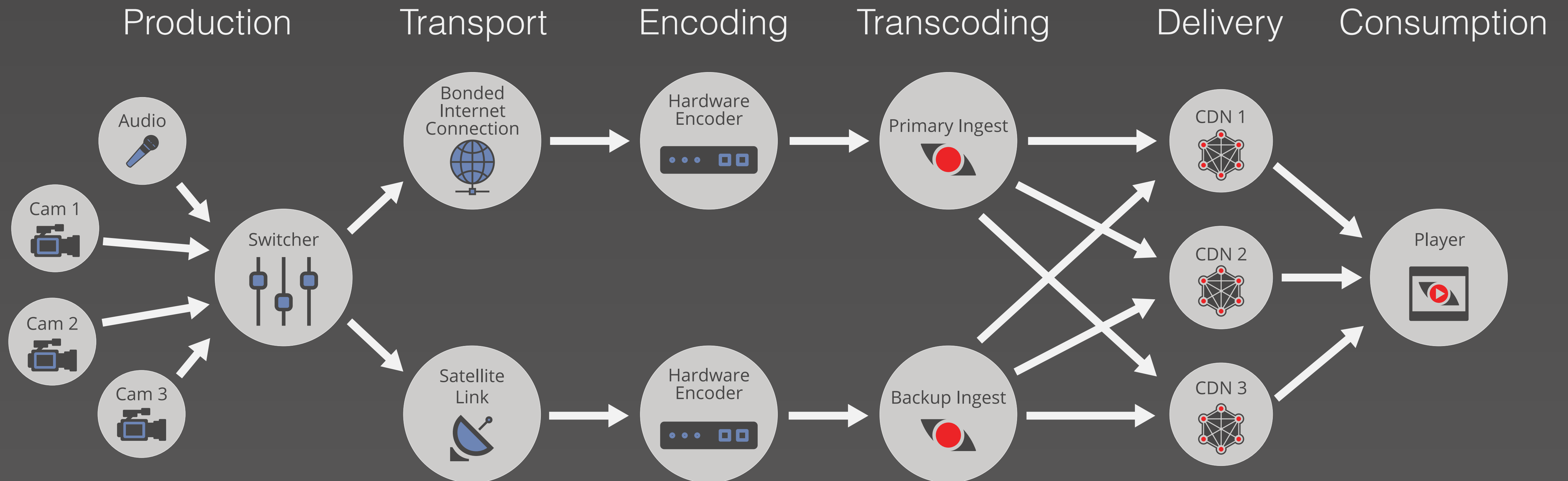
VIEW THIS ADVERTISEMENT AT YOUR OWN RISK



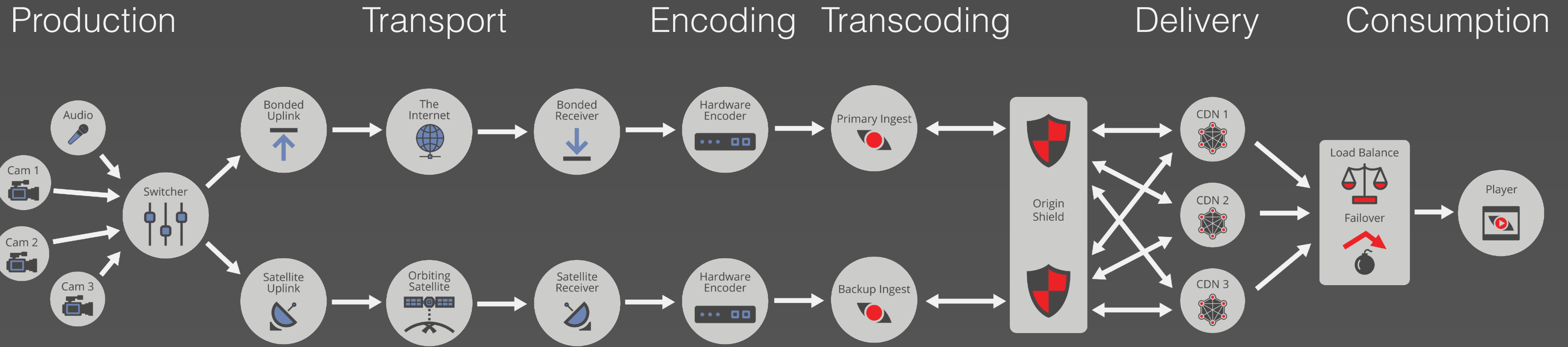
Live Streaming: Don't

If you aren't shitting your pants every time you go live, you don't know what you are doing

“Simple” 15 Step Process



Ok... Actual 21 Step Process

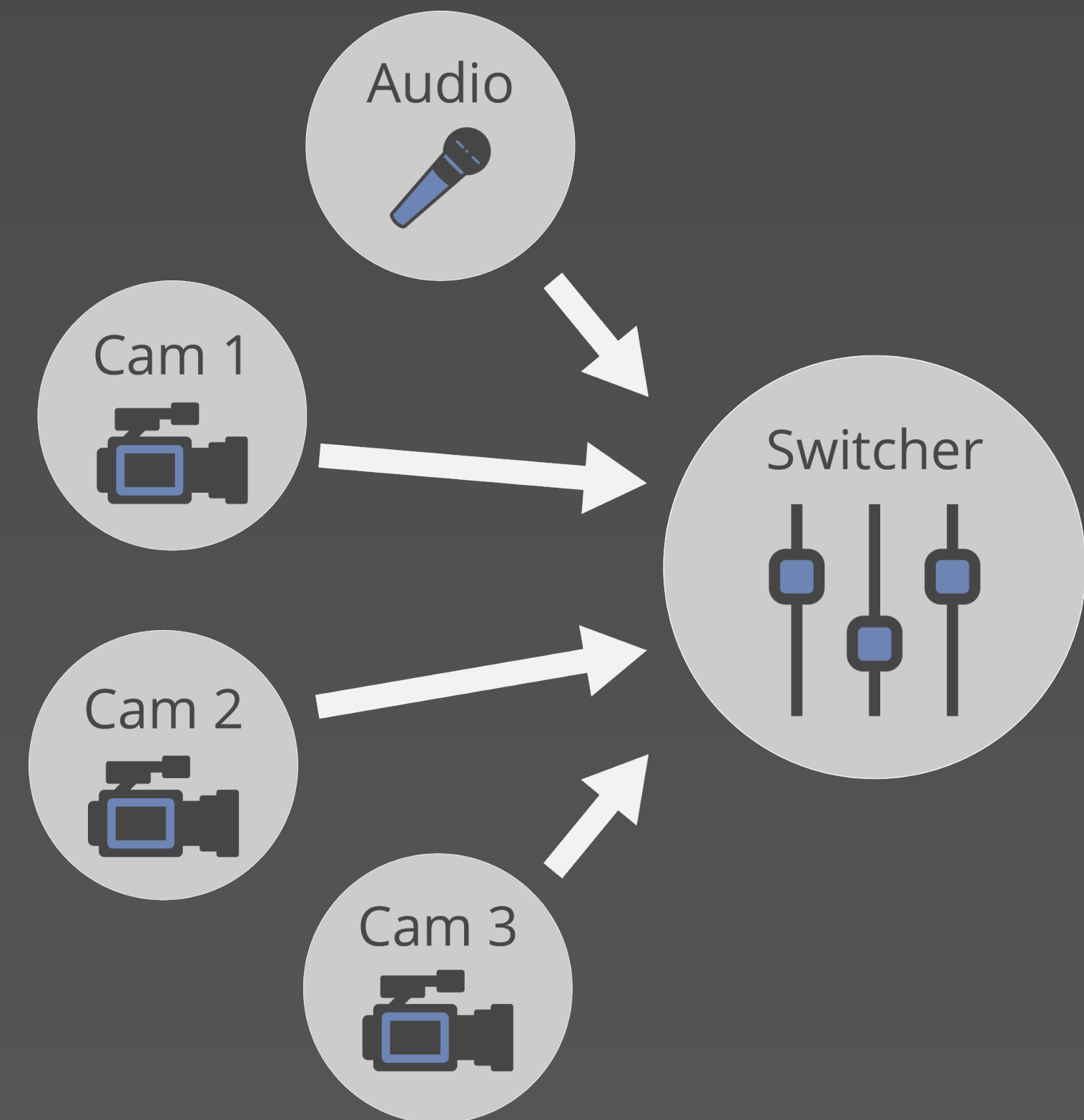


Signal Flow Overview

- Camera / Switcher / production / audio
- Contribution / Uplink (+ on site connectivity)
- Ingest and Transcoder (“Origin”)
- CDN (“Edge”) & DNS
- Player / Player Engines / embed
- 3rd party destinations/targets (e.g. FB Live)

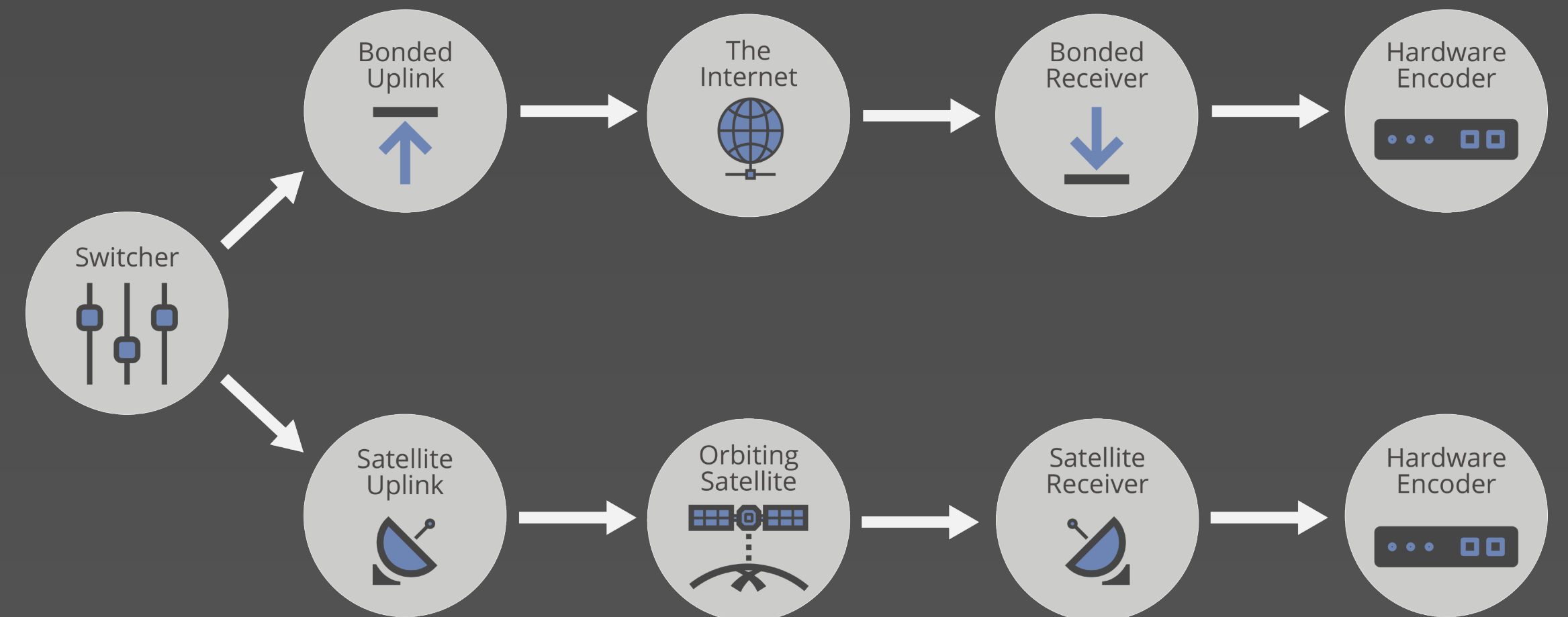
Camera / Switcher / production / audio

- Production gear fails
 - Have 2 of everything
 - I would rather have 2 cheap video switchers rather than a single expensive one
- Production teams fail
 - Guys pulling out audio cables while we press the go live button
 - Wrong/faulty cable used - audio out of phase (garbled audio on mono)



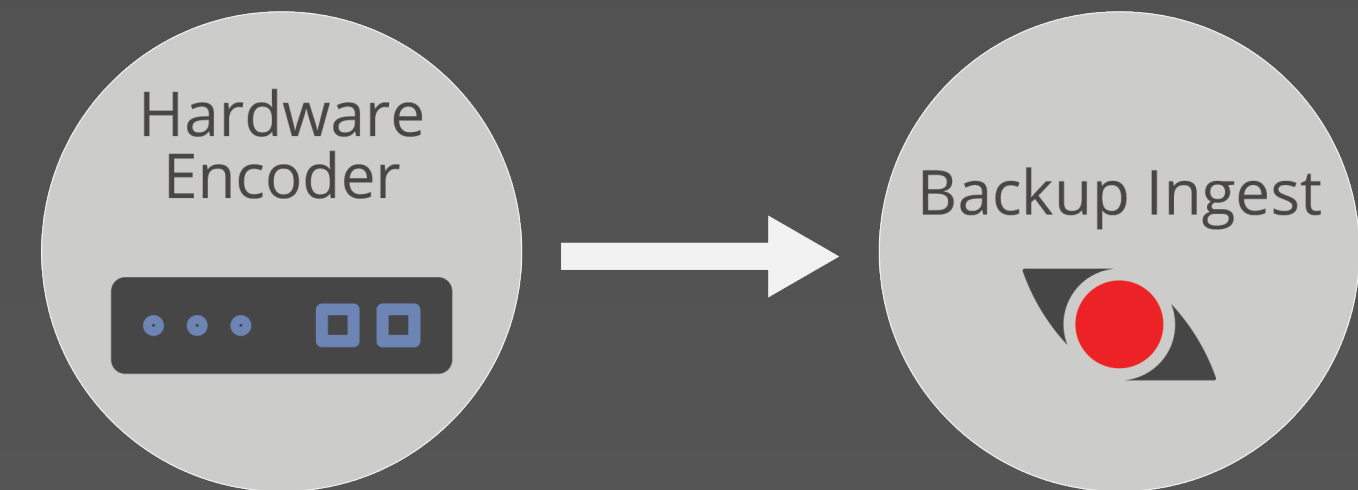
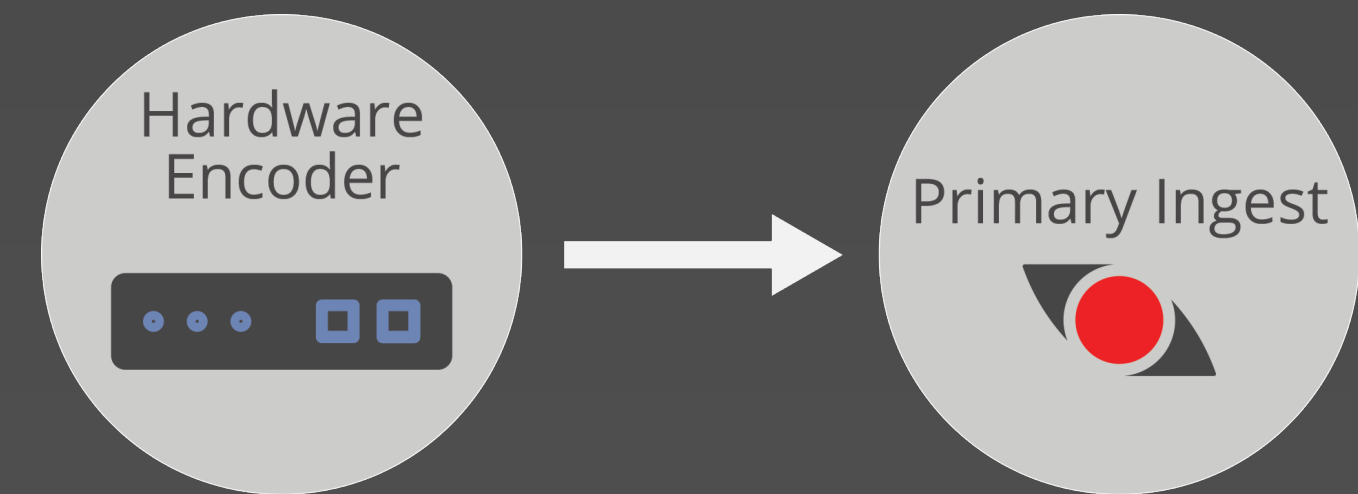
Contribution / Uplink (+ on site connectivity)

- Use reliable contribution encoders
 - Software solutions are fine but wont work for something you care about
 - For the love of god, don't use a phone
- Use diverse signal paths
 - Fiber + sat + onsite internet + bonded cellular solution etc
 - Never trust the venue internet



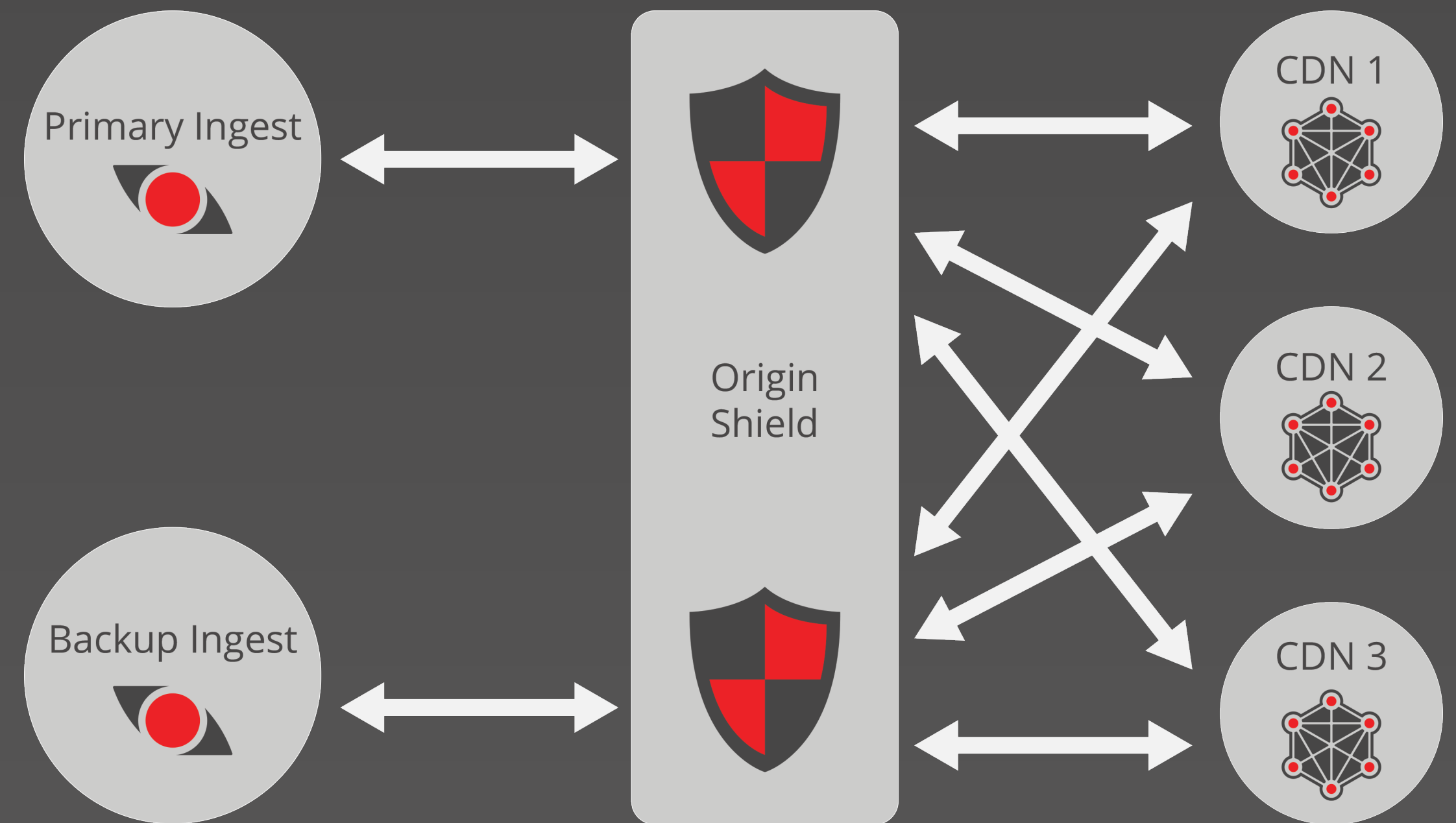
Ingest and Transcoder (“Origin”)

- Stream to Primary and Backup ingest as a minimum
- Stream to both simultaneously over redundant and diverse network paths
- Input streams ideally are key frame / time code aligned so transcoded output is identical on both primary and backup
- Transcode and packetize to HLS / Dash for maximum compatibility
- Don't forget low bitrate / audio only streams for low bandwidth consumers
- Name segments & set sane cache headers on your playlist and segments to prevent stale content



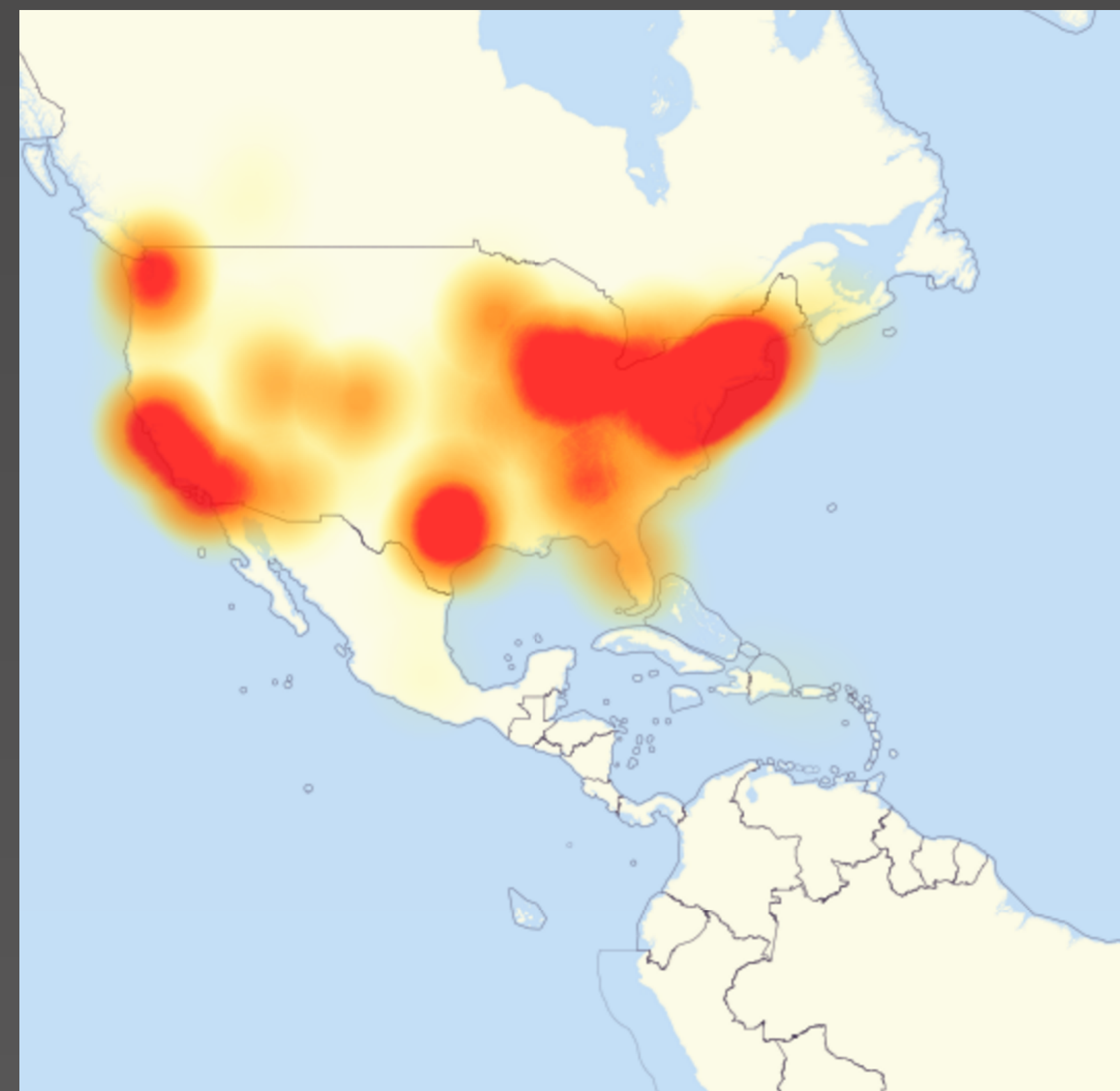
CDN (“Edge”)

- Using a single CDN is a recipe for failure. If this hasn't blown up in your face yet, it will
- Go with a Multi-CDN strategy. If this is too hard to manage yourself (integration, legal, pricing) use a one stop shop Multi-CDN provider
- Where is your audience? What peaks do you expect in what regions? Ask your CDN provider what their edge capacity / max concurrents. Let them know in advance if you expect big spikes
- Consider an intermediary caching layer or origin shield so the CDN(s) don't overwhelm your origin



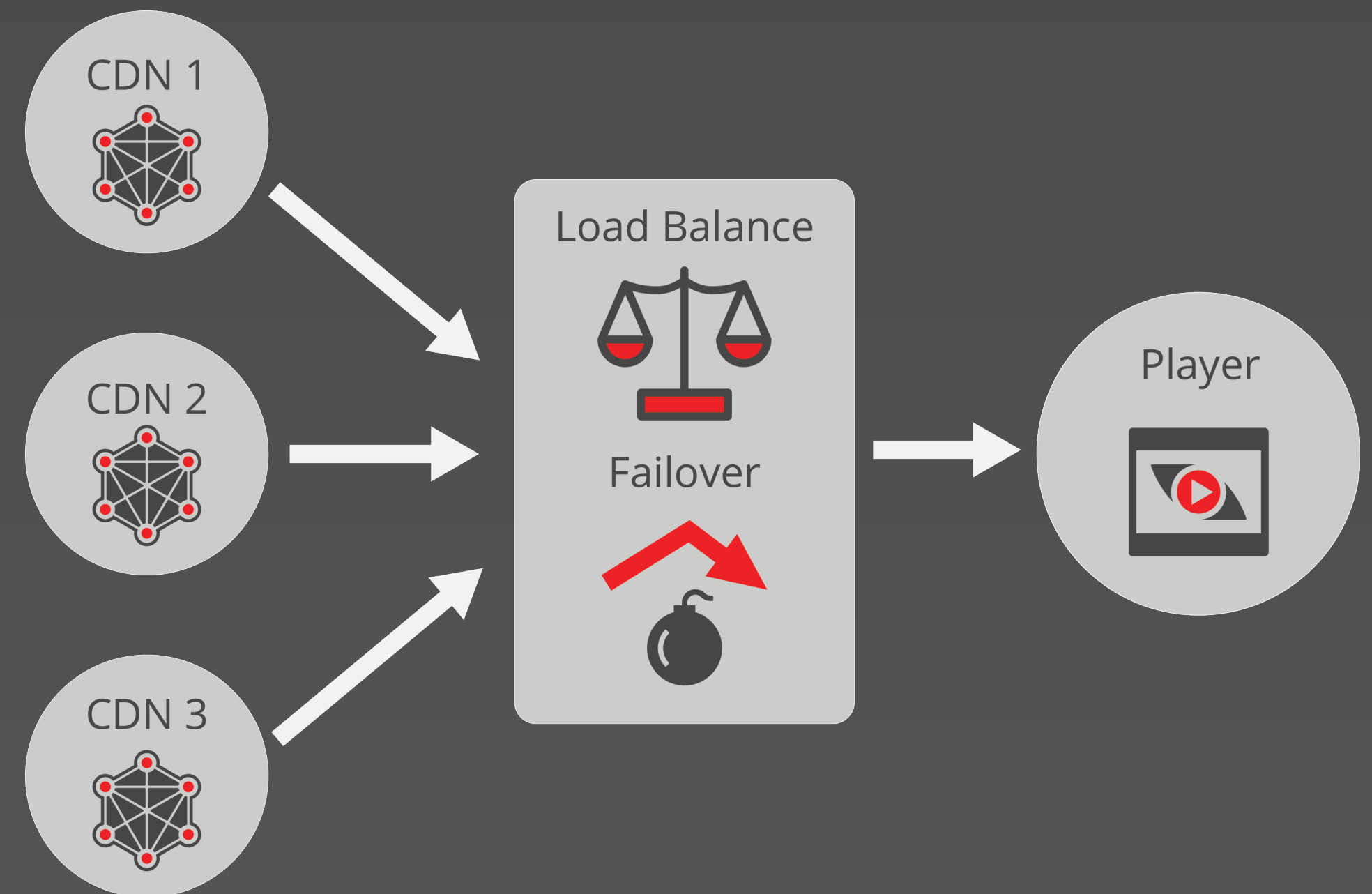
DNS

- DDoS attacks escalating - don't become collateral damage
- Choose your primary DNS wisely
- Invest in redundant DNS infrastructure
- If your DNS is simple (eg no fancy traffic management) and static having a backup/ slave is easy - export your zone file and import it into another provider
- Master/slave for simple setups (automated)
- Dual primary for high end setups - use API to mirror your setup



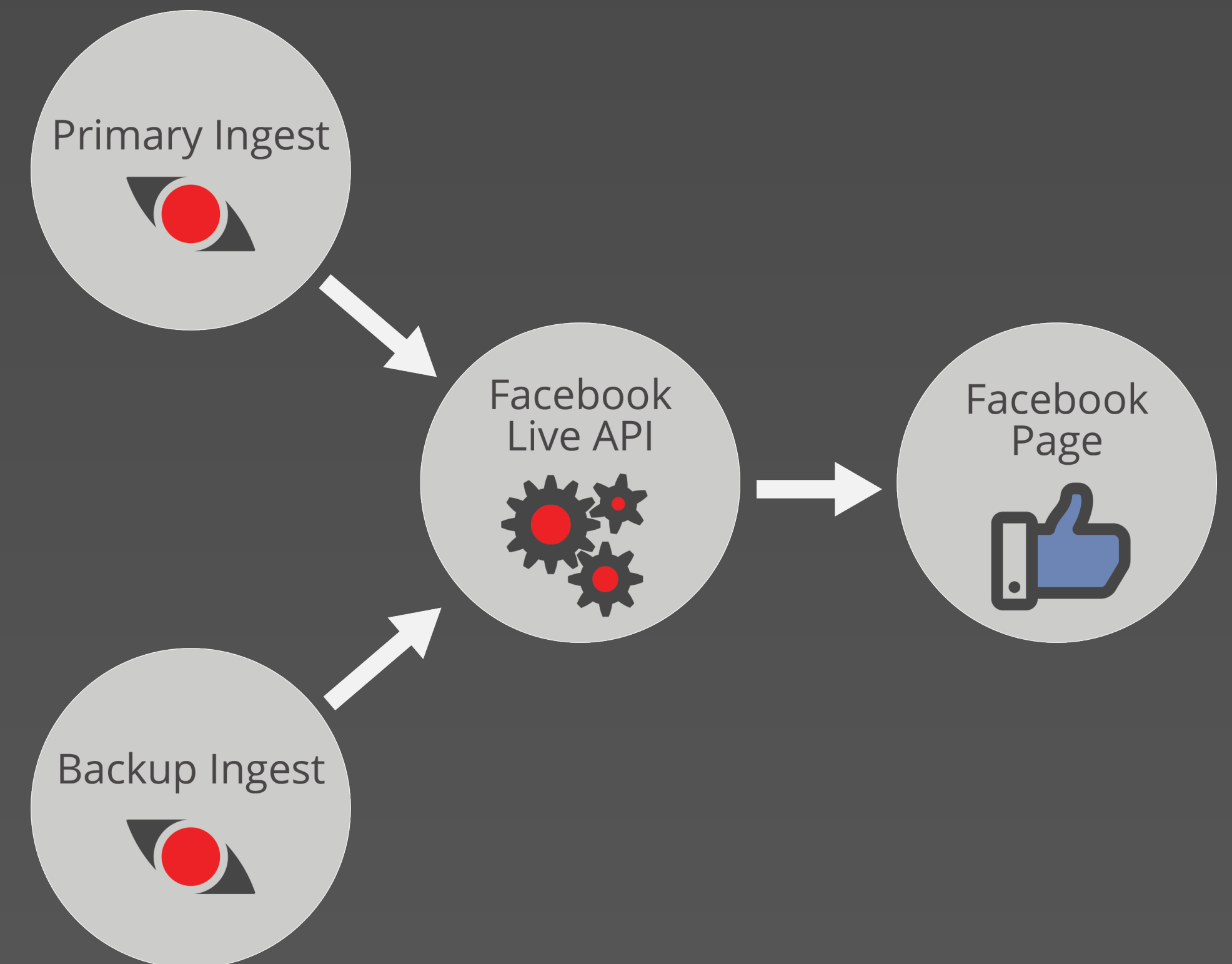
Player / Player Engines / embed

- Both HLS and DASH have protocol level support for multiple sources representing the same content, however not all playback engines honour it.
- Load balancing and failover is often best implemented at the player level.
- Smart Players can select best performing source before playback starts
- Player can failover on client side - often preferable / more responsive than DNS
- Using a hosted player vendor - ask what their CDN & DNS arch is. Failover/redundant?



3rd party destinations/targets (e.g. FB Live)

- Reasons to go live on Facebook
 - Sports
 - Breaking news
 - Known ending, but unknown timeframe (buzzfeed watermelon)
 - Interactivity
 - Authentic look at something rare



Best practice for Facebook Live

- Use pro equipment and workflows if you care about quality (no phones!)
- Be on the money with your stream starts/stops so archive/VoD looks good
- Leverage the stream preview function to check quality before going live
- Send in best possible contribution - 720p 4000Kbit/s 2 second keyframe
- If you stop pushing to ingest (e.g. your primary encoder is down) you have < 3 minutes to resume streaming before your stream 'ends' (manual failover)
- “Cleanly” end your streams so viewers aren’t left hanging (see above).

Thanks! Questions? Epic Fail Stories?

Eric Zawolkow, Media Systems Engineer - Facebook

James Broberg, Founder, CEO - StreamShark.io (james.broberg@streamshark.io)