



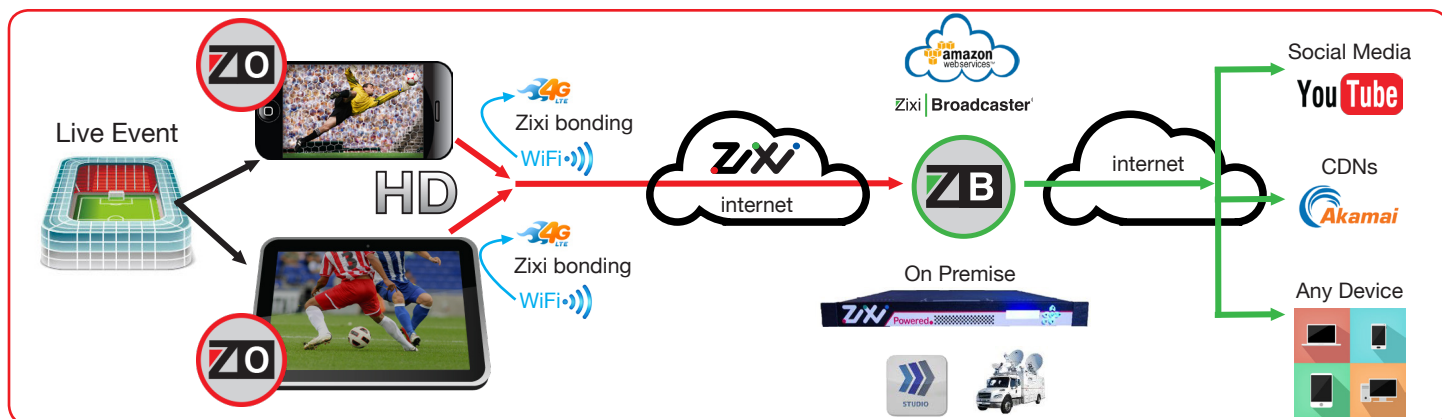
OnAir



Secure reliable contribution from any mobile device.

Zixi | OnAir™ provides a mobile entry point to the Zixi Internet video transport architecture allowing users to stream live from a mobile phone or tablet. Zixi's On-Air software app can run on IOS/Android devices and uses the built in camera to encapsulate the encoded video and audio and prepare them for transmission over standard Internet connections using Zixi's UDP-based transmission protocols.

Zixi's Transport Stream architecture deploys monitoring, control, and intelligence along the entire transmission path. It is uniquely designed to deliver the quality and reliability that enables live contribution from the any mobile device to a Zixi cloud or on premise Broadcaster. Zixi On-Air can also bond 3G/4G and WiFi connections to send the highest bit rate possible. This capability delivers a high quality stream even in degraded network conditions. Zixi's On-Air app is ideal for live event producers and ENG workflows (sports/news) who are looking to capture a breaking news story or alternative views of concerts, sports events, theater that can be used in the live broadcast or as cutaways throughout the event.



Zixi | OnAir

Major Benefits

1. Zixi On-Air uses the mobile HW encoder providing significant improvement in video quality
2. **Network bonding** – combining WiFi and 4G simultaneously for better reliability and higher bandwidth transmission
3. **Encoder feedback** – controlling the encoder bitrate according the highest available bandwidth. (iOS only)
4. Zixi protocol ensures the highest quality transmission with low latency (sub-sec latency)

Specs

- **Android devices**
 - Support streaming from an external camera (GoPro/Webcam) while using the mobile device for encoding and streaming to the public network
- **iOS devices**
 - Encode at full HD up to 1080p60
 - HW encoder feedback
- **Input**
 - Internal Mobile Device Camera
 - External Mobile Device Camera Though USB (Android only)
- **Output**
 - Zixi Protected Transport Protocol