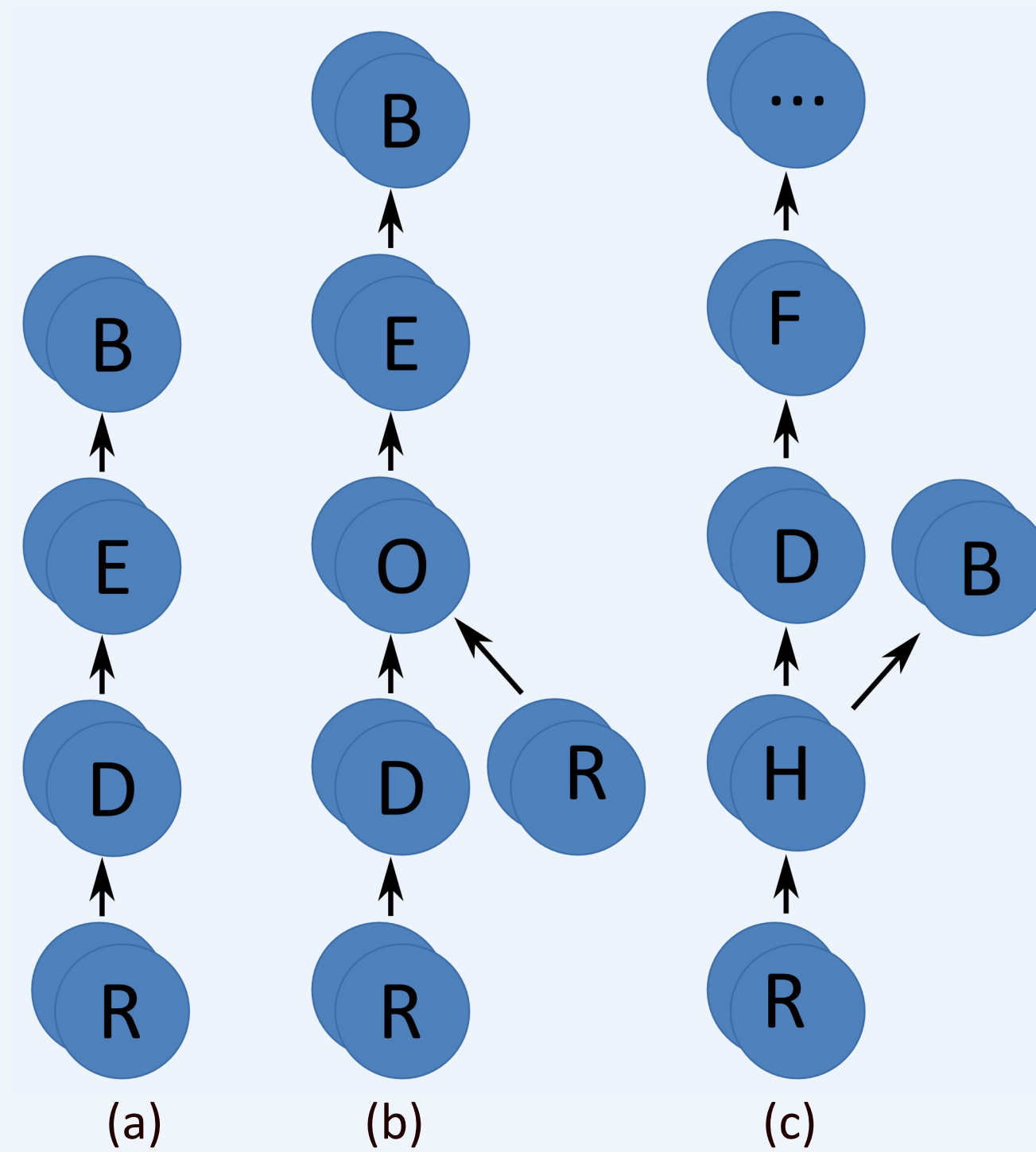


Motivation and Contributions

- More and more **inexpensive IP-enabled video-recording devices** (e.g. smartphones, tablets, ...)
- Many possible **use cases** for these devices, e.g.
 - Live broadcasting of user content
 - Lectures and Talks in Academia
 - CCTV systems
- **Automated real-time analysis and mining** requires large-scale systems due to sheer amount of video material
- **Contributions**
 - The Livescale Toolkit for building video processing dataflows inside Nephele
 - Components and architecture for scalable real-time video processing

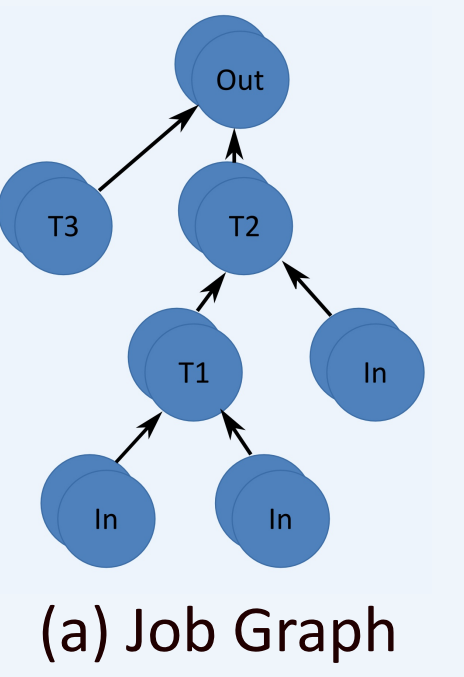
The Livescale Toolkit

- Reusable **building blocks for real-time video and audio stream processing** in Nephele
 - Reusable code for writing Nephele tasks
 - Data model (video frames, audio samples, metadata, etc)
 - Integration of third-party libraries, e.g. ffmpeg
 - Available as open-source (see [1])
- Extensible sets of predefined tasks
 - Receivers and Broadcasters
 - Decoders and Encoders
 - Frame Processors
 - Hubs and Routers
- Job graphs can be created by **mixing pre- and user-defined tasks**. Examples:
 - (a) Transcoding job
 - (b) Adding content overlays
 - (c) Concurrent stream broadcast and analysis

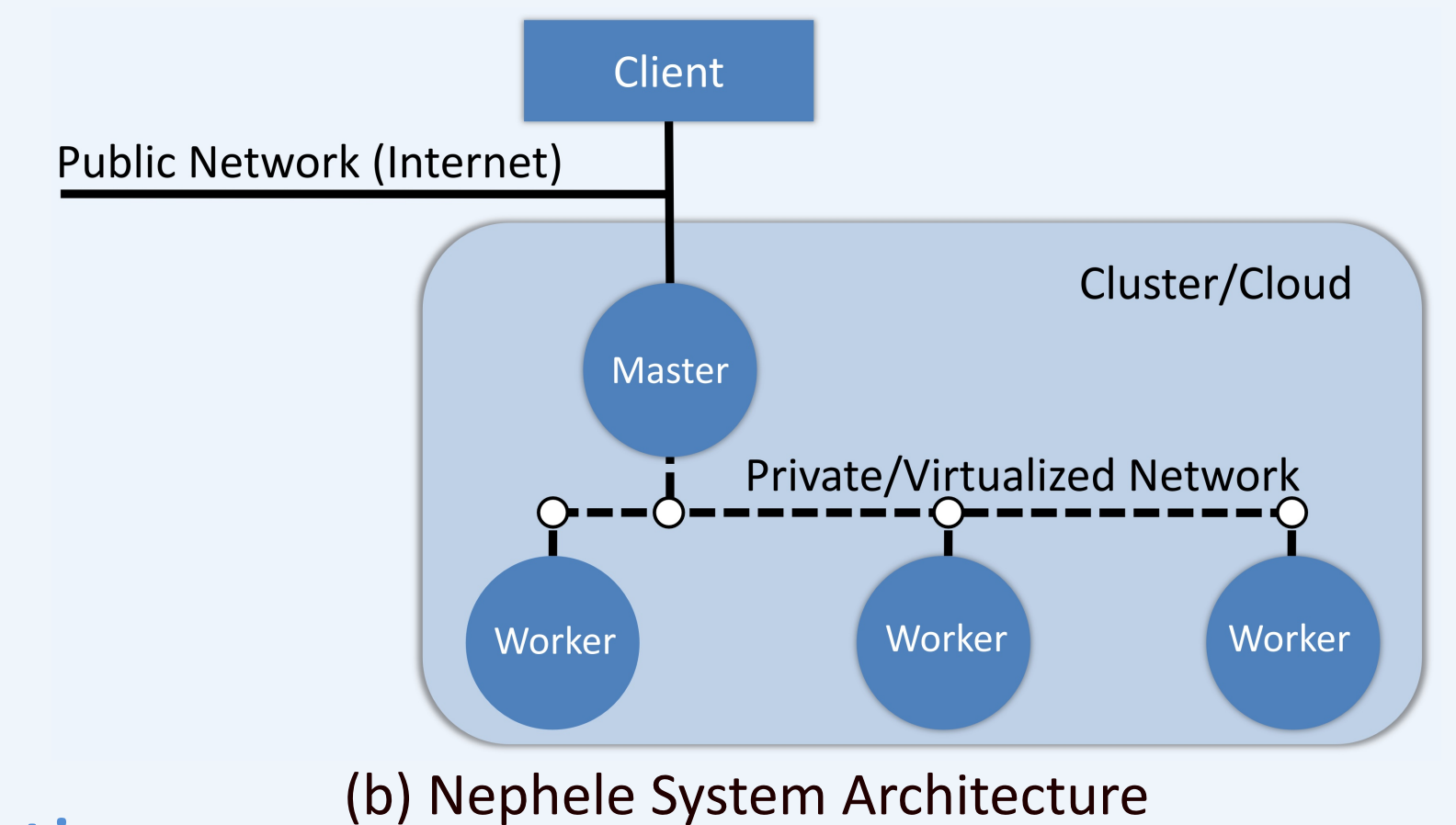


Scalable Dataflows in Nephele

- Nephele [3] executes **directed acyclic graphs** ("job graphs") on clusters and clouds
 - Available as open-source (see [2])
 - Vertices run as threads that execute user-defined functions (UDFs)
 - Edges are message-oriented channels (network or in-memory)

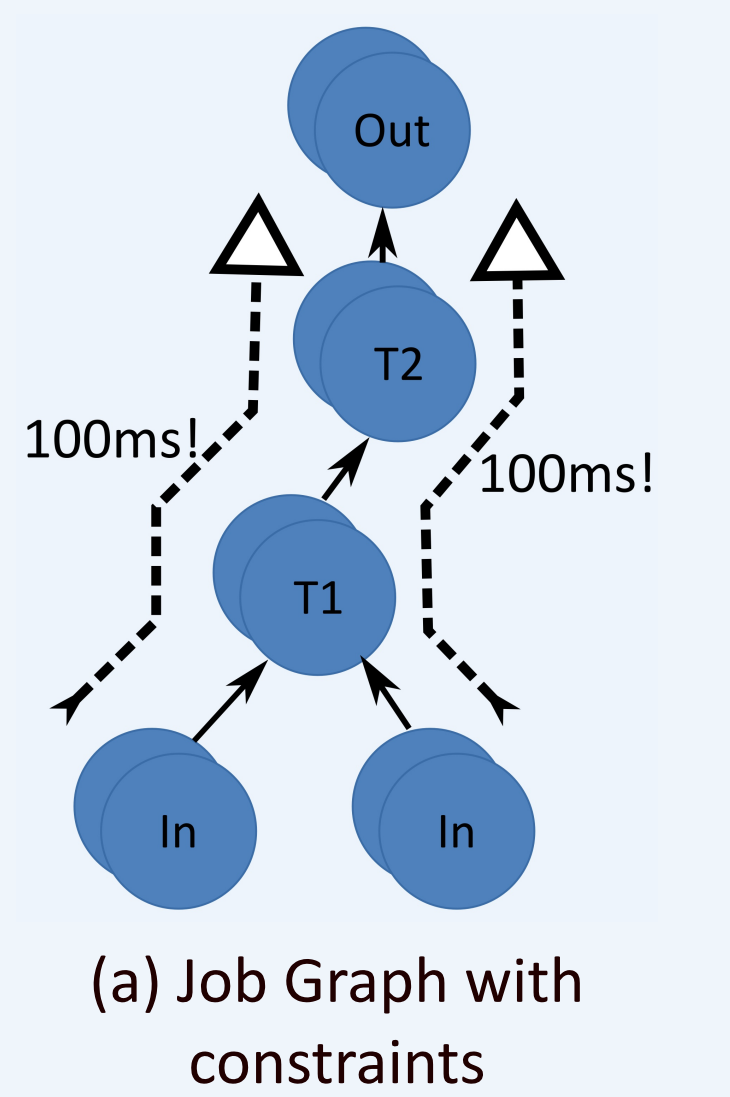


- Nephele has **master-worker architecture**
 - Master provides scheduling and coordination
 - Workers execute tasks



- Nephele **manages major aspects of distributed computation**
 - Data-parallel and pipeline-parallel execution of sequential UDFs
 - Task scheduling and resource management
 - Fault tolerance

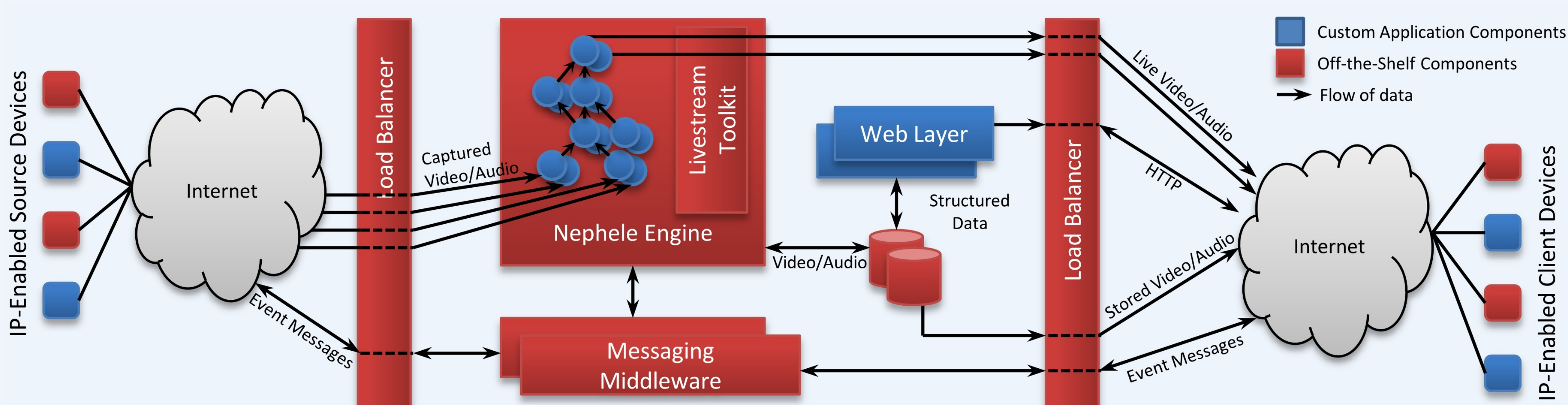
- Nephele supports **latency constraints** (see [4])
 - User can define latency constraints on the job graph
 - Nephele trades off latency and throughput to fulfill constraints



Demonstration

- **Sample application using the Livescale Toolkit**
 - Groups video streams by tags
 - Ranks streams in group by quality
 - Clients choose the best quality stream from desired groups
- Application **includes Android-based app** for sending and watching streams
 - Available as open-source, see [1]
- **Follows general architecture** outlined below

Real-Time Video Processing Architecture



- **Optional components**
 - Messaging middleware for coordination
 - Web Layer for user management etc
 - Scalable persistent storage outside Nephele
- **Mixes off-the-shelf and custom components**

References

- [1] The Nephele Livescale Toolkit: <http://www.github.com/bjoernlohrmann/livescale-toolkit>
- [2] The Stratosphere Streaming Distribution: <http://www.github.com/bjoernlohrmann/stratosphere>
- [3] D. Warneke, and O. Kao, "Exploiting dynamic resource allocation for efficient parallel data processing in the cloud," IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 6, pp. 985-997, IEEE, 2011
- [4] B. Lohrmann, D. Warneke, and O. Kao, "Nephele Streaming: Stream processing under QoS constraints at scale," Cluster Computing, Springer, 2013

