Media redirection for Spice remote computing solution

Optimizing media stream processing for media players and VoIP clients in virtual desktop infrastructures

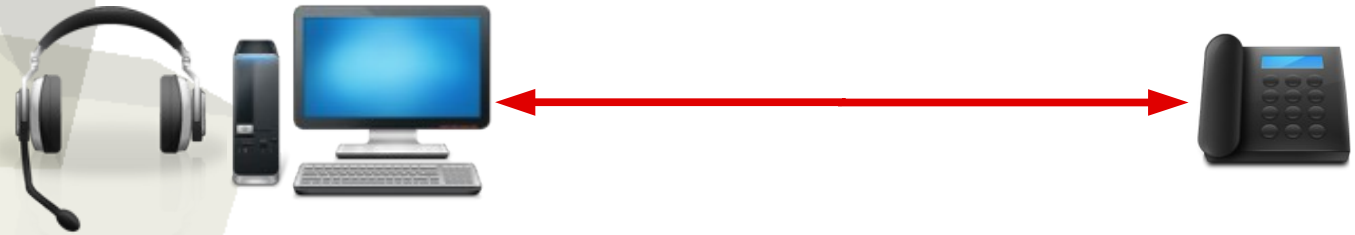
Lyakhov F.A., Mazur E.S., Kuzin A.M., Semenov S.K.

Common media processing usecases

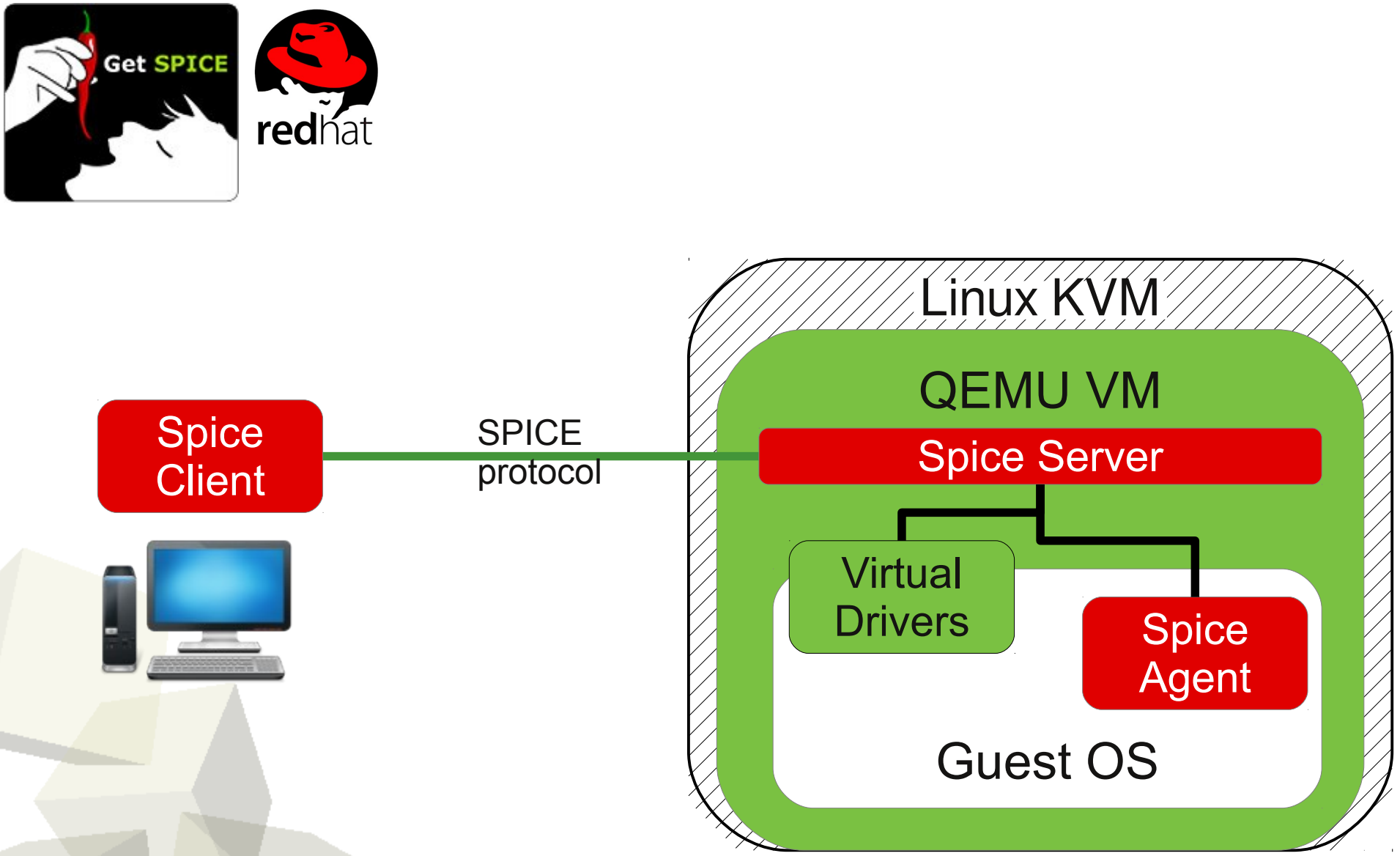
- Creating and playing of local media content (a file)

- Playback of remote media content

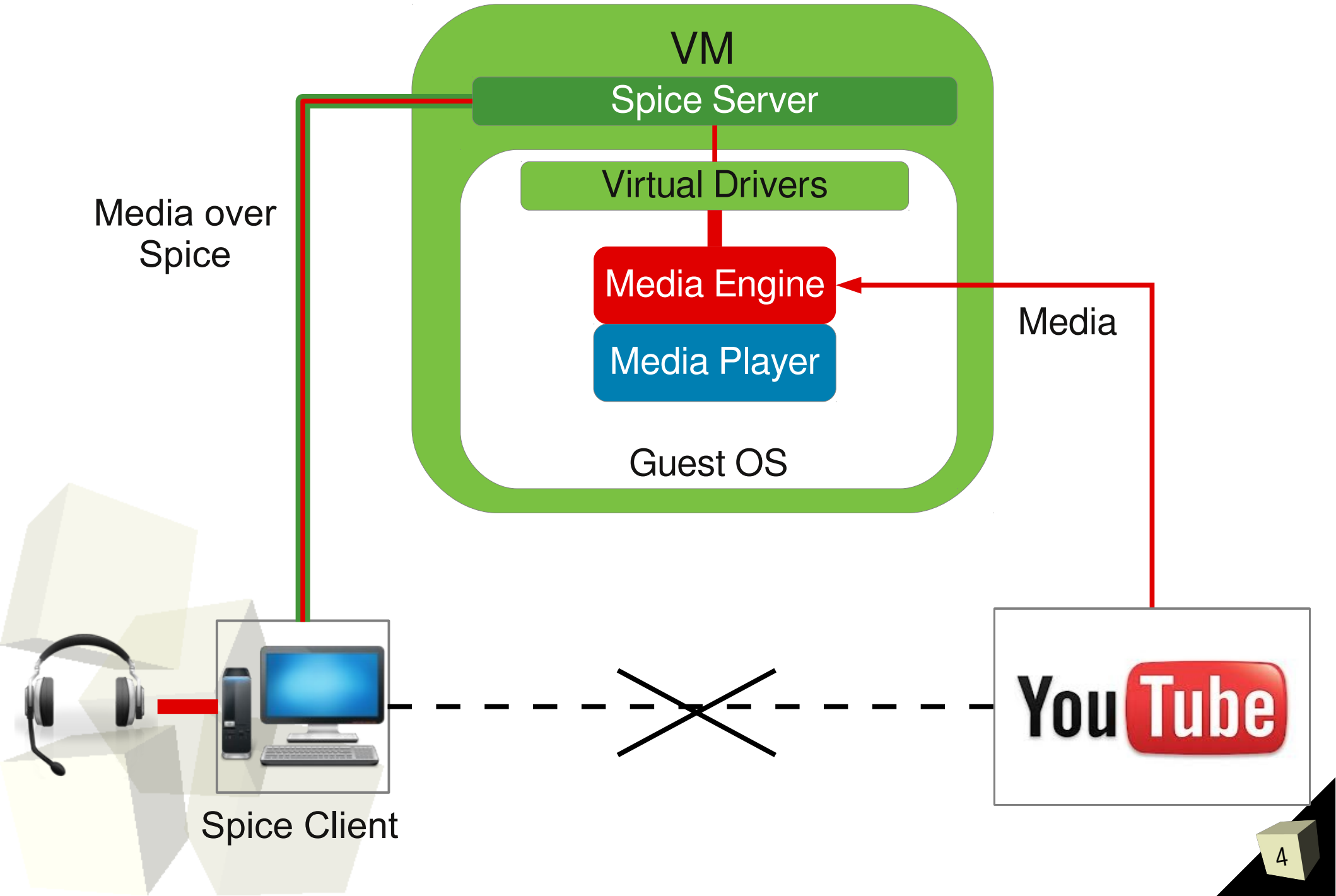
- Telecommunications



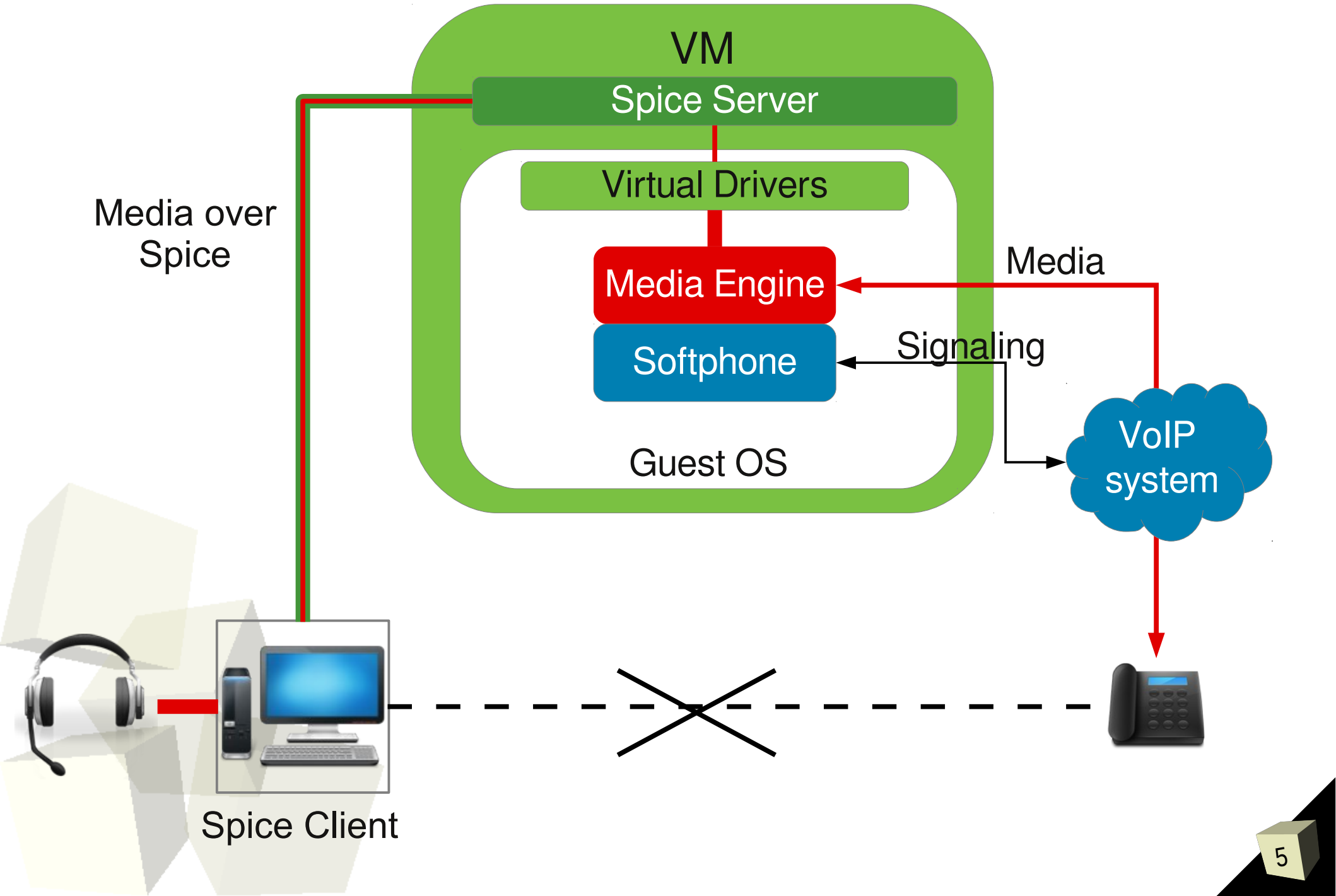
Red Hat Spice overview



Playback of remote media in VDI



VoIP-system in VDI



Problem definition

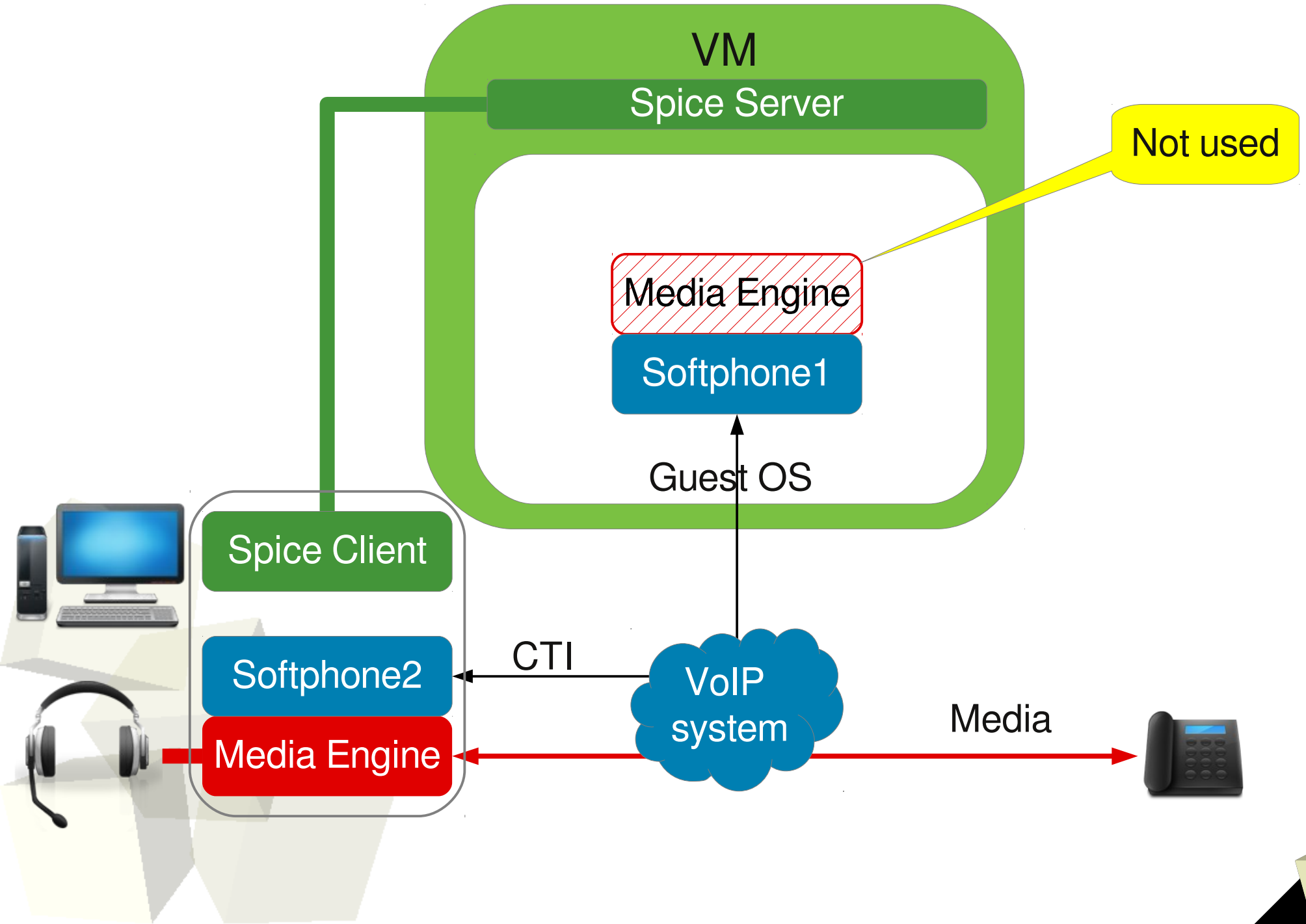
Hair-pinning effect:

- Media streams are passing through the virtualization server, not peer-to-peer
- Media streams are transcoded at the server

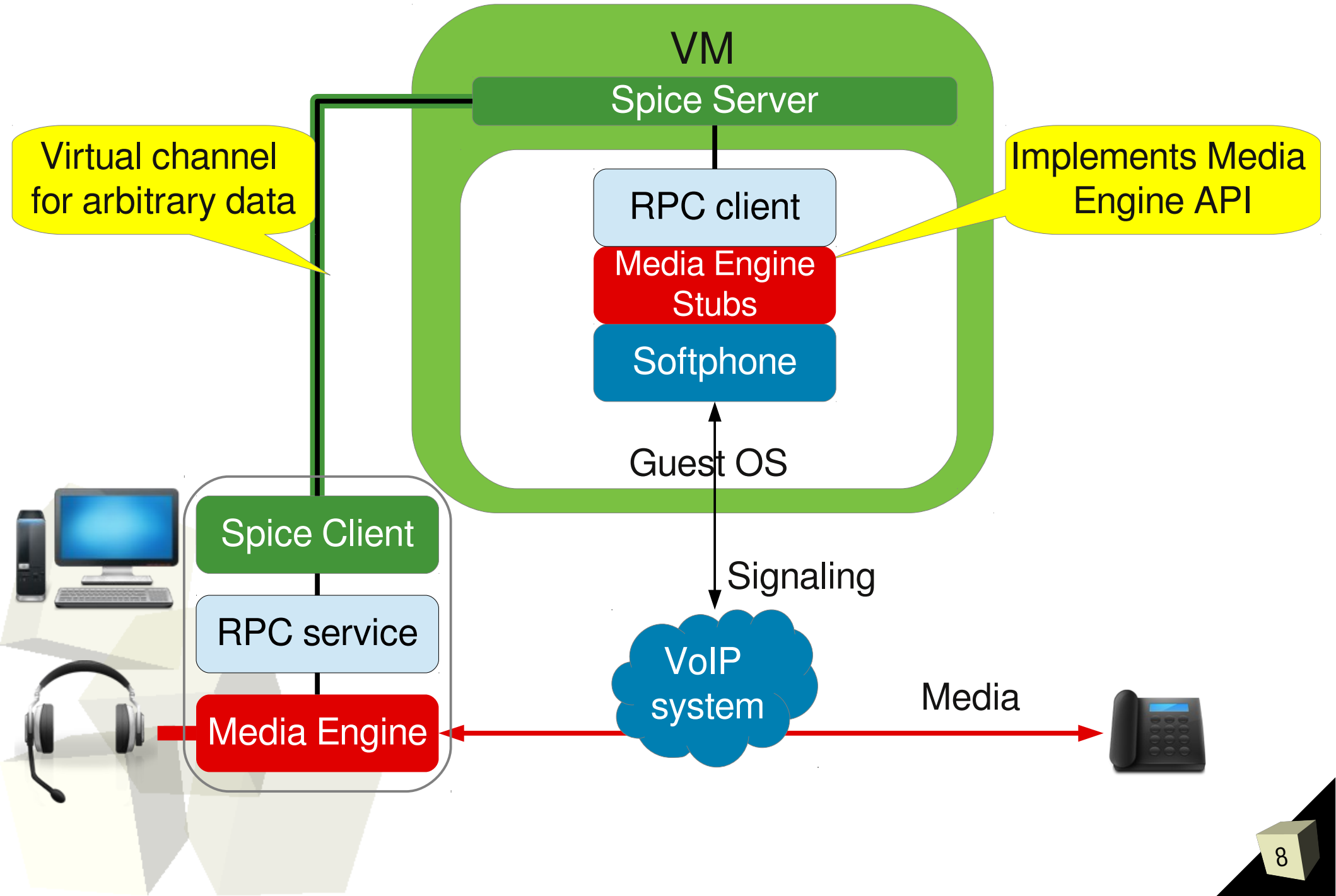
This results in

- Increased network load
- Increased server CPU load, less VM density
- Increased latency, jitter, packet loss
- Possible quality loss of media streams

Custom VoIP solution



Media redirection concept



Media redirection concept: video specifics

VM

Spice Server

RPC client

Media Engine Stubs

Softphone

Guest OS

Overlay Renderer

Spice Client

RPC service

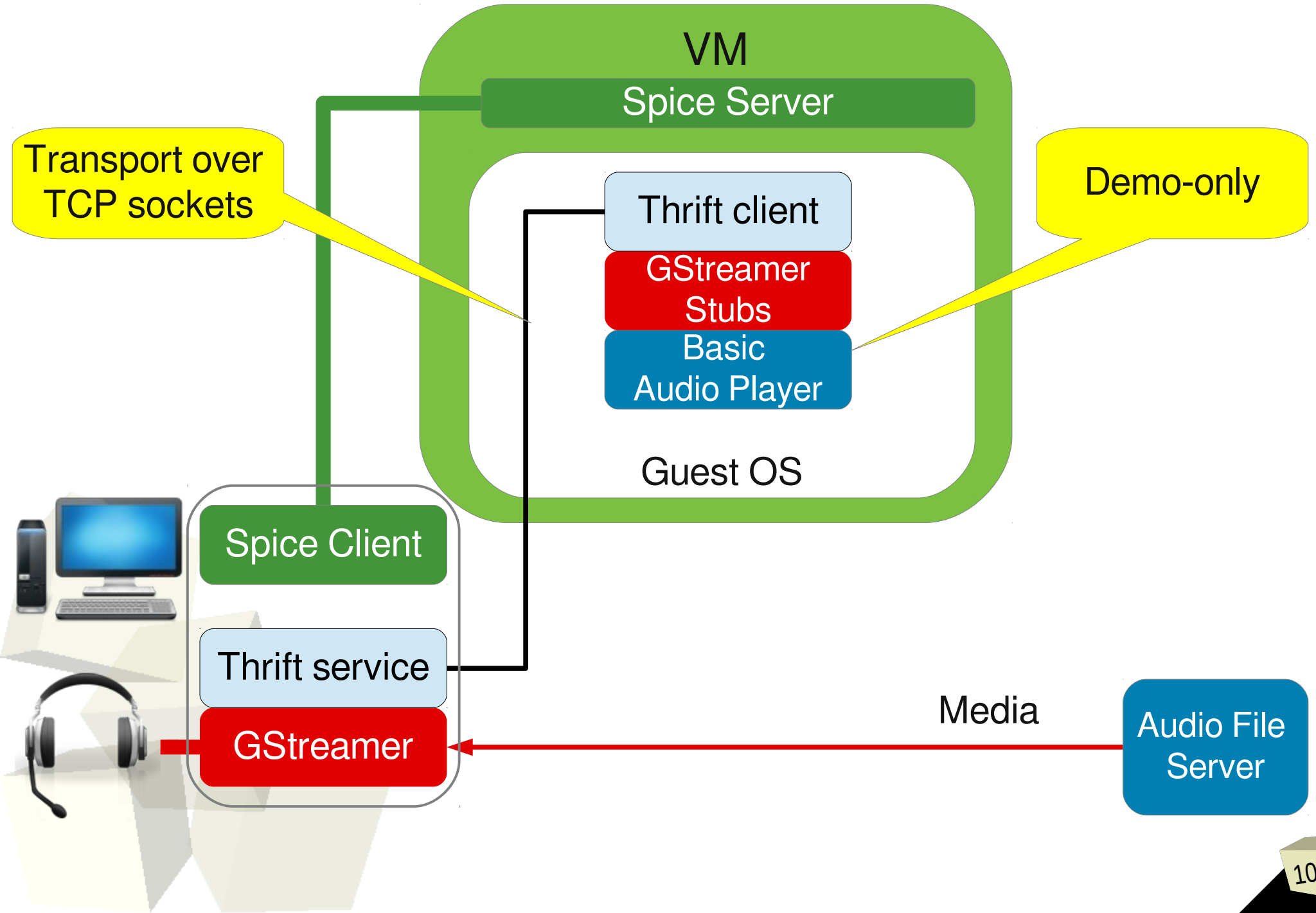
Media Engine

Signaling

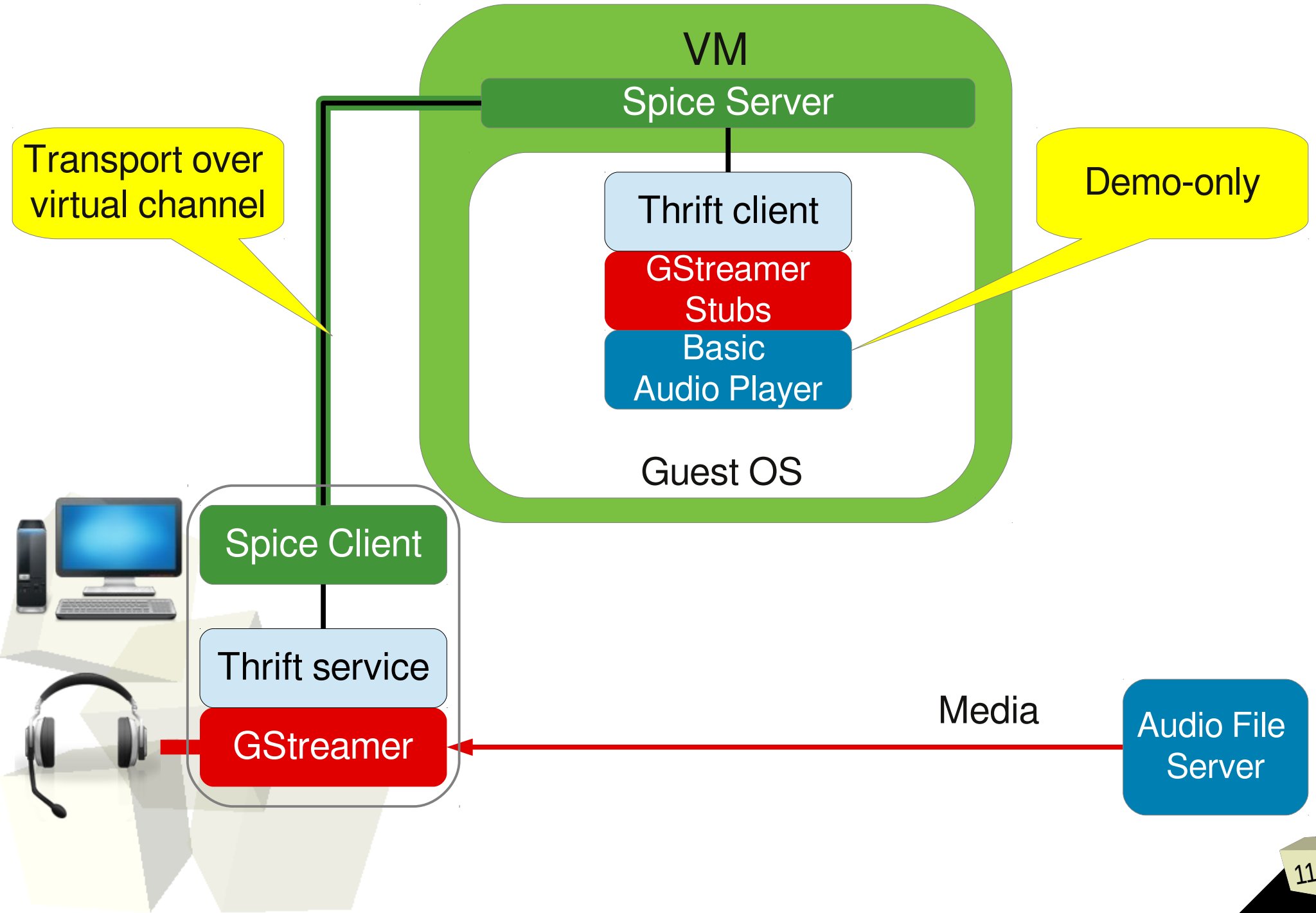
VoIP system

Media

Media redirection prototype v0.1



Media redirection prototype v0.2



Feature evolution plan

- Prototype v0.1:
  - Basic demo-only audio player streaming audio file from a server
  - Thrift RPC over basic TCP sockets
  - Minimum GStreamer API implemented via RPC
- Prototype v0.2: Thrift RPC over a channel in Spice
- Prototype v0.3: Demo-only softphone
- Version 1.x: real-world audio player and softphone
- Version 2.x: overlay renderer for Spice, video

Architecture & design considerations

- Component-based universal design:
  - Media Redirection should allow implementing support for other remote computing systems
  - RPC system as Spice extension, not nailed down
- RPC system choice:
  - Apache Thrift with custom transport
  - Google Protocol Buffers marshaling with custom RPC and transport
- Support for multiple common Media Engines:
  - GStreamer
  - VLC
  - Google Media Engine

New Spice API

- API for virtual channels
  - At Guest OS: shared/static library for apps
  - At Spice Client: plug-in interface for services
  - Multiple apps can connect to one service
  - Initiation of connection: as D-Bus services
- Overlay Rendering API

Fault-tolerance discussion topics

- Spice Client disconnect / crash:
  - Keep services running e.g. keep audio path of an IP call, allow reconnect
  - Freeze Guest application till re-connect?
- Need for client application with UI to control services e.g. stop the audio in case virtualization server became unavailable
- RPC service and Media Engine crash recovery – separate processes?
- If Media Engine is in a separate process, how it will be rendering video into Spice Client window?
- Migration support

Conclusion

- Media processing problem in VDI described
- Media Redirection for Red Hat Spice proposed with component design and re-use of technologies
- Prototype demonstrated based on
  - Apache Thrift for RPC
  - GStreamer for Media Engine

Thanks for watching! Q&A