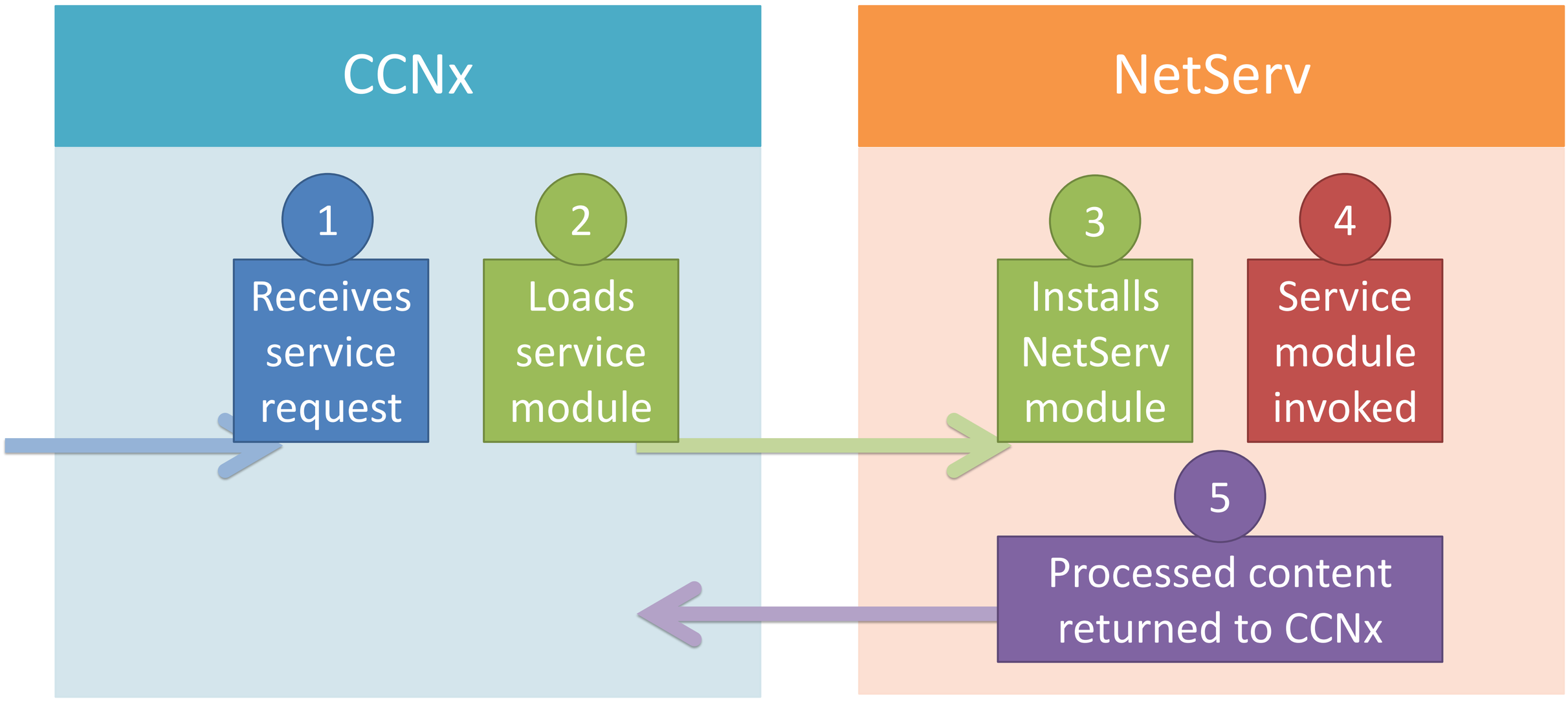
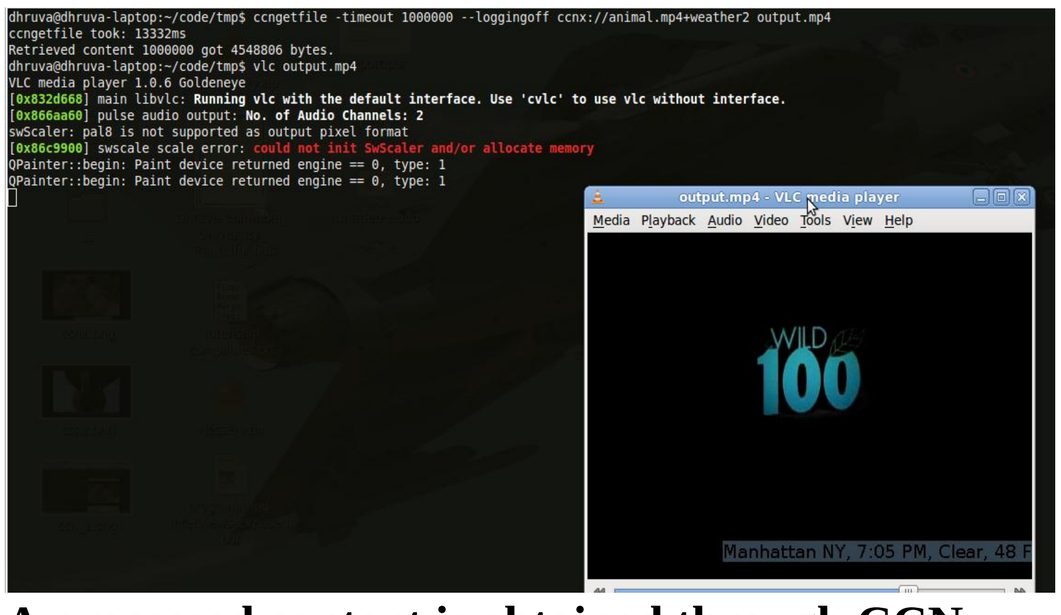
Dynamic Service Scalability in Information-Centric Networks

Suman Srinivasan [1], Dhruva Batni [1], Volker Hilt [2] and Henning Schulzrinne [1]
[1] *Columbia University,* [2] *Alcatel-Lucent*
sumans@cs.columbia.edu, dlb2155@columbia.edu, volker.hilt@alcatel-lucent.com, hgs@cs.columbia.edu

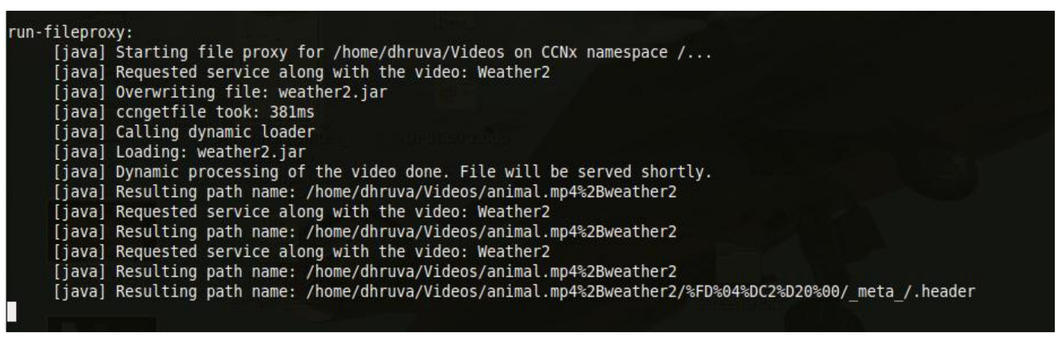


The architecture for our CCNx dynamic service scalability implementation. When CCNx receives a content request that it interprets as a service, it loads the service module (over CCNx), processes the content using the module in NetServ, and then publishes it in CCNx space.

Screenshots



A processed content is obtained through CCNx and played in VLC media player. There is a small watermark with weather information at the bottom right of the video, showing that this is processed video obtained from CCNx.



When a service request is made, the service module (in addition to the content) is downloaded dynamically from CCNx, and invoked on the content.

Pseudo-Code for Implementation

```
ccnName = "ccnx://test.mpg+weather";
service, file = parse (ccnName);
download ("ccnx://netserv/" +
                    service + ".jar");
download ("ccnx://content/" + file);
loadLocalJAR (service + ".jar");
processedFile = loadClass(service).
       getMethod("run").invoke(file);
putFileIntoCCNx (processedFile);
```

Currently: Our implementation interprets the service request, dynamically loads the module, invokes the service module on the original content, and publishes it back into CCNx space.

Future Work: Integrate this completely with our NetServ service virtualization platform.

Conclusion: We have a working prototype of a system that allows service processing in CCNx, thus allowing for dynamic service scalability.