Decentralised Content Management Protocol for Pervasive Displays

1. What are the design implications for creating a rural pervasive display network with decentralised content management?
2. What improvements to the effectiveness of disseminating information in a display network can be gained when designing it in a decentralised manner?

Previous Work:
Showboater
P-Layers
StoryBank

Written In Python

Developed with Kanban

Using a distributed hash table and arranging all the pervasive displays in a star network mitigates performance issues and creates data redundancy.

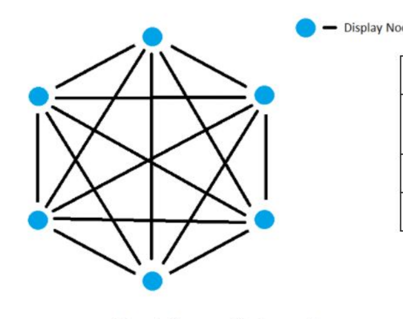
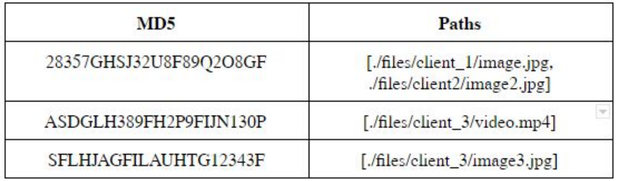


Figure 2 - Star network for data transfer



MD5	Paths
28357GHSJ32U8F89Q2O8GF	[./files/client_1/image.jpg, ./files/client2/image2.jpg]
ASDGLH389FH2P9FIJN130P	[./files/client_3/video.mp4]
SFLHJAGFILAUHTG12343F	[./files/client_3/image3.jpg]

Table 2 - example meta_data hash table

Files are synchronised, and are transferred on a separate layer of the protocol.

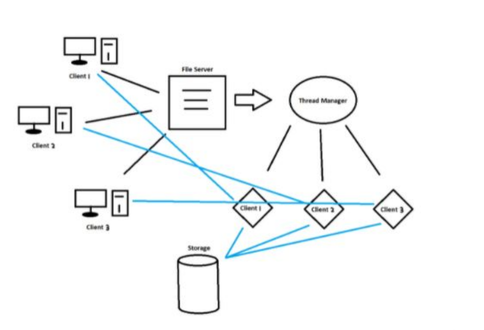


Figure 3 - Thread management of the file management level

Displaying the content is achieved with a HTTP server.

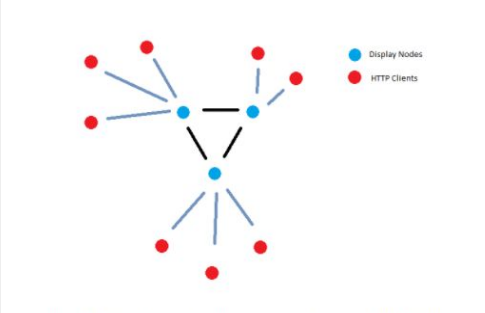
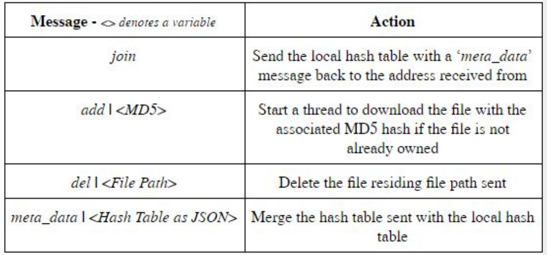


Figure 4 - Networked nodes with optional local networks of HTTP clients

- Any device can be used to view content.
- Break down into two tiers of display network.
- Allows customisation with any web app.



Message - *<> denotes a variable*	**Action**
join	Send the local hash table with a '*meta_data*' message back to the address received from
add | *<MD5>*	Start a thread to download the file with the associated MD5 hash if the file is not already owned
del | *<File Path>*	Delete the file residing file path sent
meta_data | *<Hash Table as JSON>*	Merge the hash table sent with the local hash table

Table 3 - Messages in the broadcast layer protocol

Only 4 protocol commands required for synchronising content & displays.

Student: Joshua Green 956213 technical project **Supervisor:** Stuart Nicholson