Hivestack Ad Server Workflow

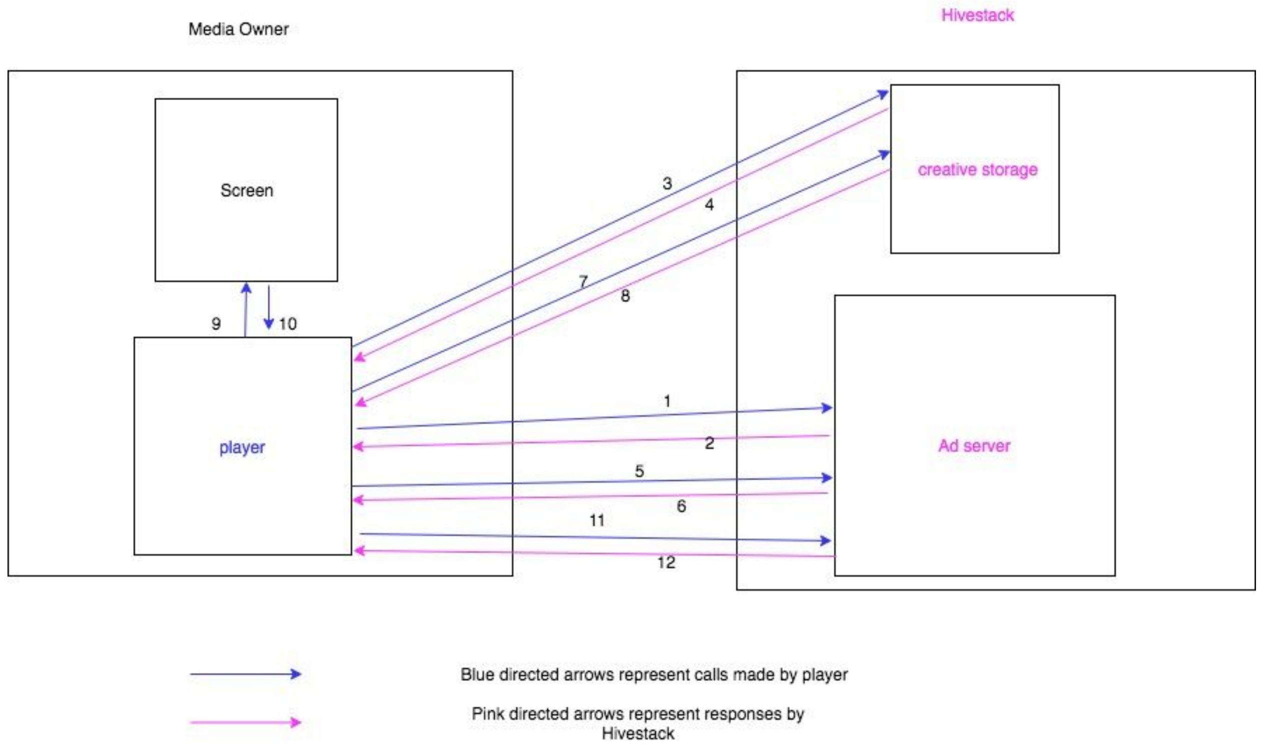
Download all possible creatives --> Ad Request --> Fetch Creative --> Play --> Confirm Play

(Download and run the project at https://github.com/Hivestack/hs-nirvana-tutorial-csharp for sample calls. You will need an app.config file with a sample screen details. If you weren't provided one, please mail at info@hivestack.com)

Terms:

1. Creative - The ad to be displayed on the Digital OOH screen
2. Screen - The Digital OOH screen that displays the creative
3. Player - The software that supports the screen to enable playing the creative

Detailed Workflow:


Media Owner
Hivestack
Screen
creative storage
9
10
3
4
7
8
player
1
2
5
6
11
12
Ad server
Blue directed arrows represent calls made by player
Pink directed arrows represent responses by Hivestack

1. The player has the option to download all the creatives it might potentially play in the next hour. The player does a GET request to the endpoint "hivestack_url/nirvana/api/v1/units/{unit_uuid}/creatives". unit_uuid is a distinct guid assigned to each screen after the inventory has been synced with Hivestack.
2. This endpoint returns all possible creatives the screen will play in the next hour. They should be downloaded and stored locally for reuse.
3. Request the creative from the creative url from the external storage. (Could be cloudfront or any other storage)
4. Save the creative locally so that they can be reused, instead of downloading them again and again. ***These steps 1 to 4 doesn't have to be done every time and it's completely optional. It's to avoid downloading the creative on the fly. Ideally, these steps 1 to 4 have to be done only once every hour.***
5. When the player finds a spot in the loop that was meant to call hivestack, It would do a GET request to the endpoint "hivestack_url/nirvana/api/v1/units/{unit_uuid}/schedulevast".
6. Ad server decides the best creative to play (depending on various factors, audience conc, impression, targeting) and returns a VAST with the information about the *creative to play* and the *confirmation url* to call if the creative was played successfully. The ad server can also return an empty VAST with 'Nothing Scheduled' if it doesn't find any good creative to play.
7. The player should parse the VAST tag (*for example, see the github project in the above url.*) and extract the creative url and the confirmation url. The player checks the creative url to check if the creative has already been downloaded and stored in the local storage. If the creative doesn't exist in the local storage then player makes the call to the creative url (usually to a storage) to get the creative.
8. The player fetches the creative from the external storage or the local storage(if cached).
9. The player sends the creative details to the screen to play.
10. The screen plays the creative.
11. The player, once the creative is played successfully calls the *confirmation url* endpoint at ad server to confirm that it has played the creative.
12. The ad server returns a '*200 OK*' with 'play confirmed' to acknowledge that it has received the play confirmation. Double confirmations will raise an error.