# Process Flow Diagram

**Swim-lane headers (top brackets):**
- Local User Data Entry
- Remote User Data Entry
- Remote (AWS)

**Legend box:**
Process flow swim-lanes
- User Interaction
- Website Query
- FPT Quest Generation Signal
- Quest Concept Generation
- Quest Implementation

## Swim-lane columns

| User Interaction | Website Query | FTP Quest Generation Signal | Quest Concept Generation | Quest Implementation |
|---|---|---|---|---|

### User Interaction
- User plays on EQemu (Eulogy-Quest)
- User wants to create a Eulogy for a fallen friend or would like to test it for further consideration.
- Use website to generate the quest? (Yes / No)
- Python direct quest initiation via aws-quest-gen-script.py

### Website Query
- User provides the name
- Optionally provide data
- Basic (tutorialb-only)?
- Prepare the FPT file

### FTP Quest Generation Signal
- Empty FTP file? (Yes / No)
- FPT with empty file: Eulogyquest_<Firstname_Lastname>.trigger (empty == famous person)
- FPT with data in the file: Eulogyquest_<Firstname_Lastname>.trigger (non-empty == user supplies information for GPT to generate a quest for a non-famous person)
- Name
- Name and Data
- Either way, we have a named file. FPT this file

### Quest Concept Generation
- Name: "First Last"
- Input → ghost-prompt-raw.py
- Output → Ghost-dialog-<First_Last>-raw.txt
- input → TODO: click-text.py → Output → TODO: ghost_task.txt (updated) OR npc<#>_task.txt
- Input → Input → get-delivery-item.py → Output → <First_Last>-delivery-item.txt
- Name: "First Last"
- Input → Input → get-delivery-target.py → Output → <First_Last>-delivery-target.txt

### Quest Implementation
- watch_triggers.py
- Listens for a specially named file Eulogyquest_<Firstname>_<Lastname>.trigger to be sent to the akk-stack/server/quests/ directory
- Automatically call on receiving a file
- Emergency Proof of Concept? (Yes / No)
- Yes → Do all, show all. None are great → create-quest.py calls base-story.py, perl-script-NPCs.py, updateNPCs.py
- No → OK! We'll have better versions of the required files
- TODO create-quest.py calls ghost-prompt-raw.py, get-delivery-item.py, get-delivery-target.py, Michael's scripts, then we call the next two scripts with better data: perl-script-NPCs.py, updateNPCs.py

**Bare Proof of Concept**
- base-story.py
- perl-script-NPCs.py
- updateNPCs.py

**EQEMU Game Server**
- Perl files scripting NPC behavior, in: server/quests/
- PEQ EQEmu database

**Proof of Concept perl file resources**
- ghost_task.txt AND npc<#>_task.txt
- <First_Last>-delivery-item.txt
- <First_Last>-delivery-target.txt

## Notes:
Currently aws-quest-gen-script.py only supports empty FTP (famous person), but you can still pass data in the FTP anyway.