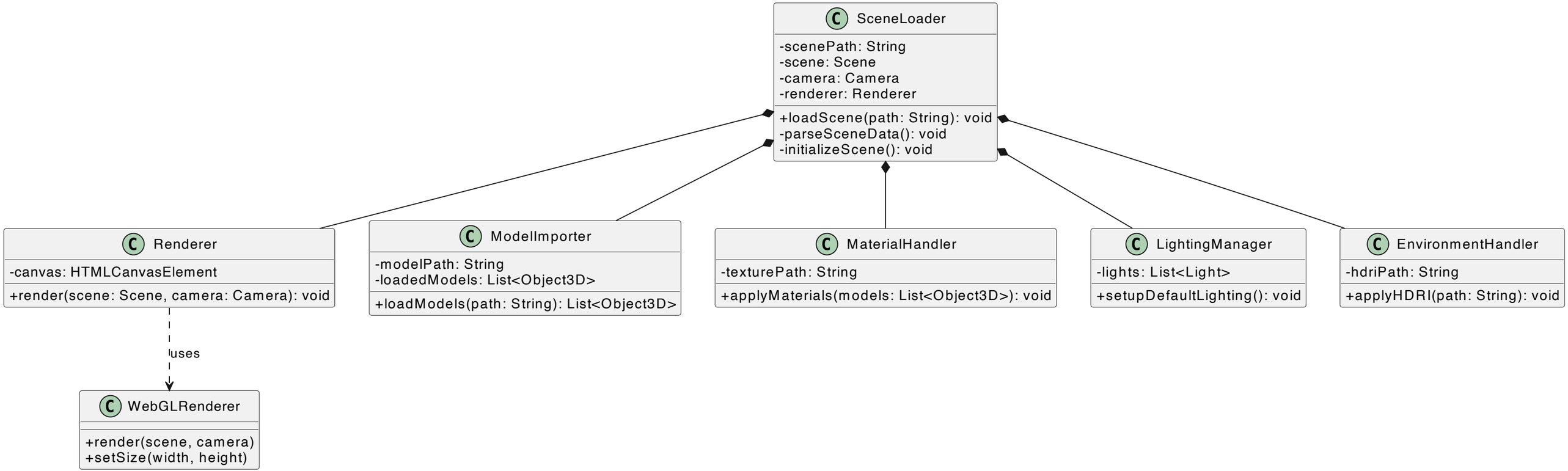
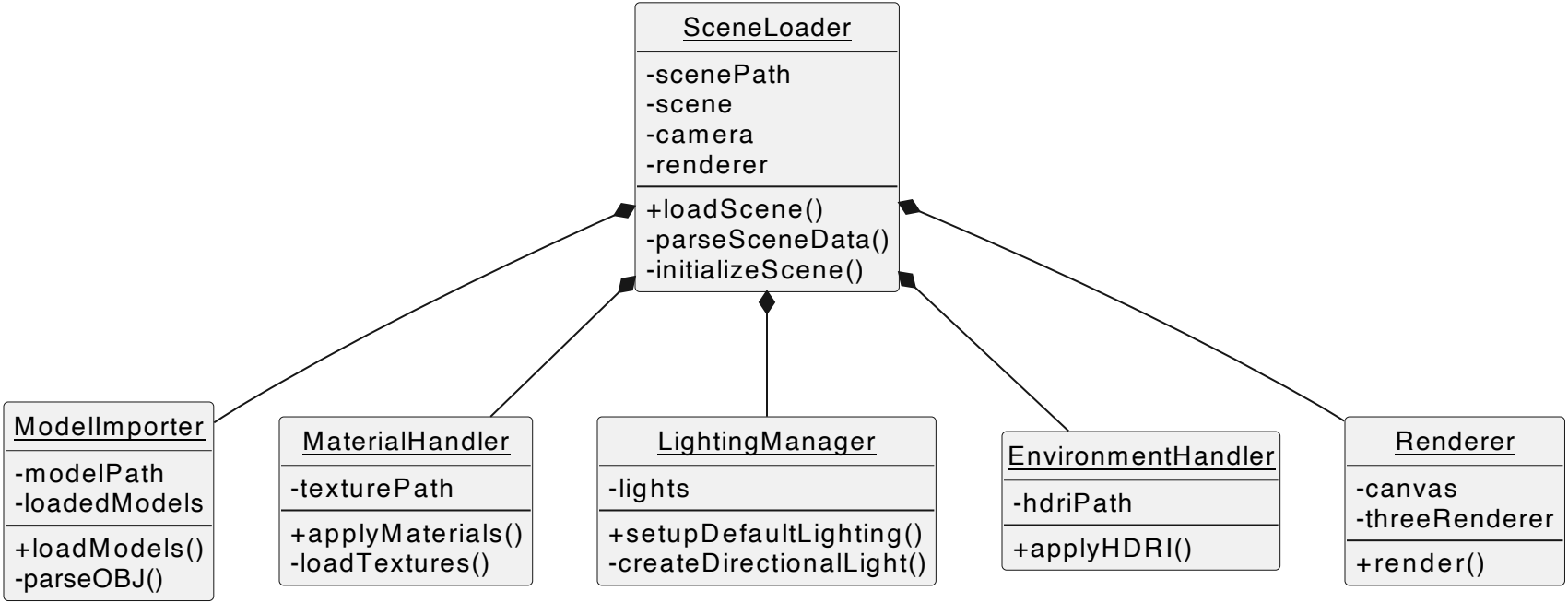
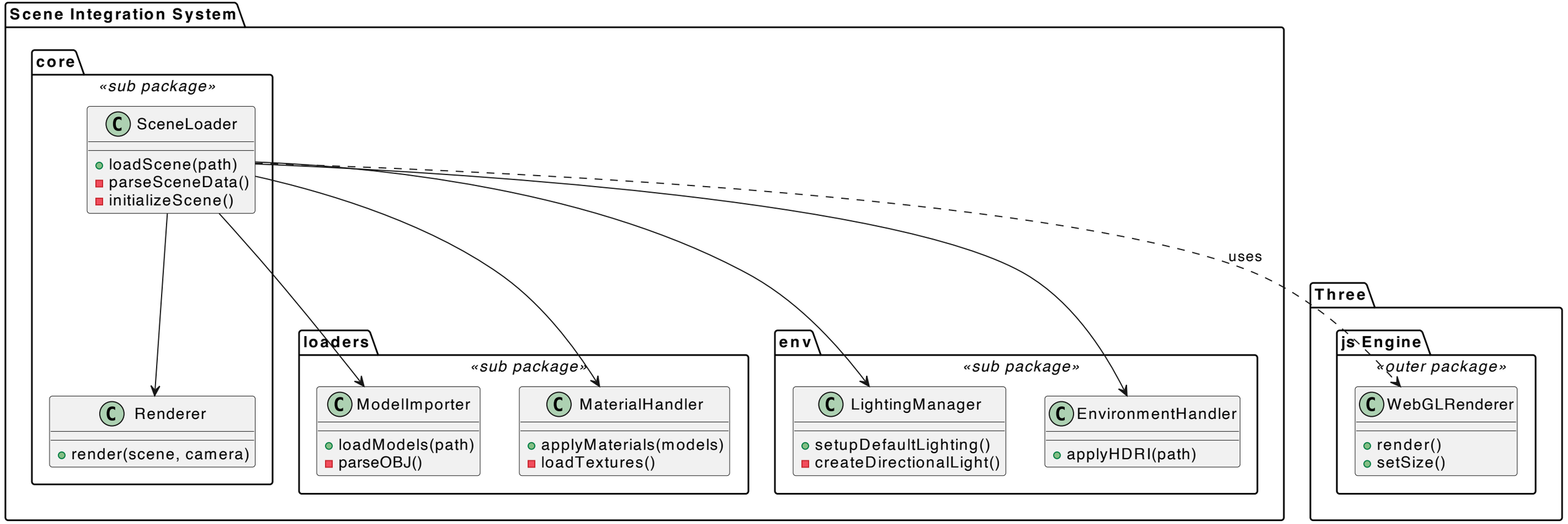
Class Diagram: Scene Integration System

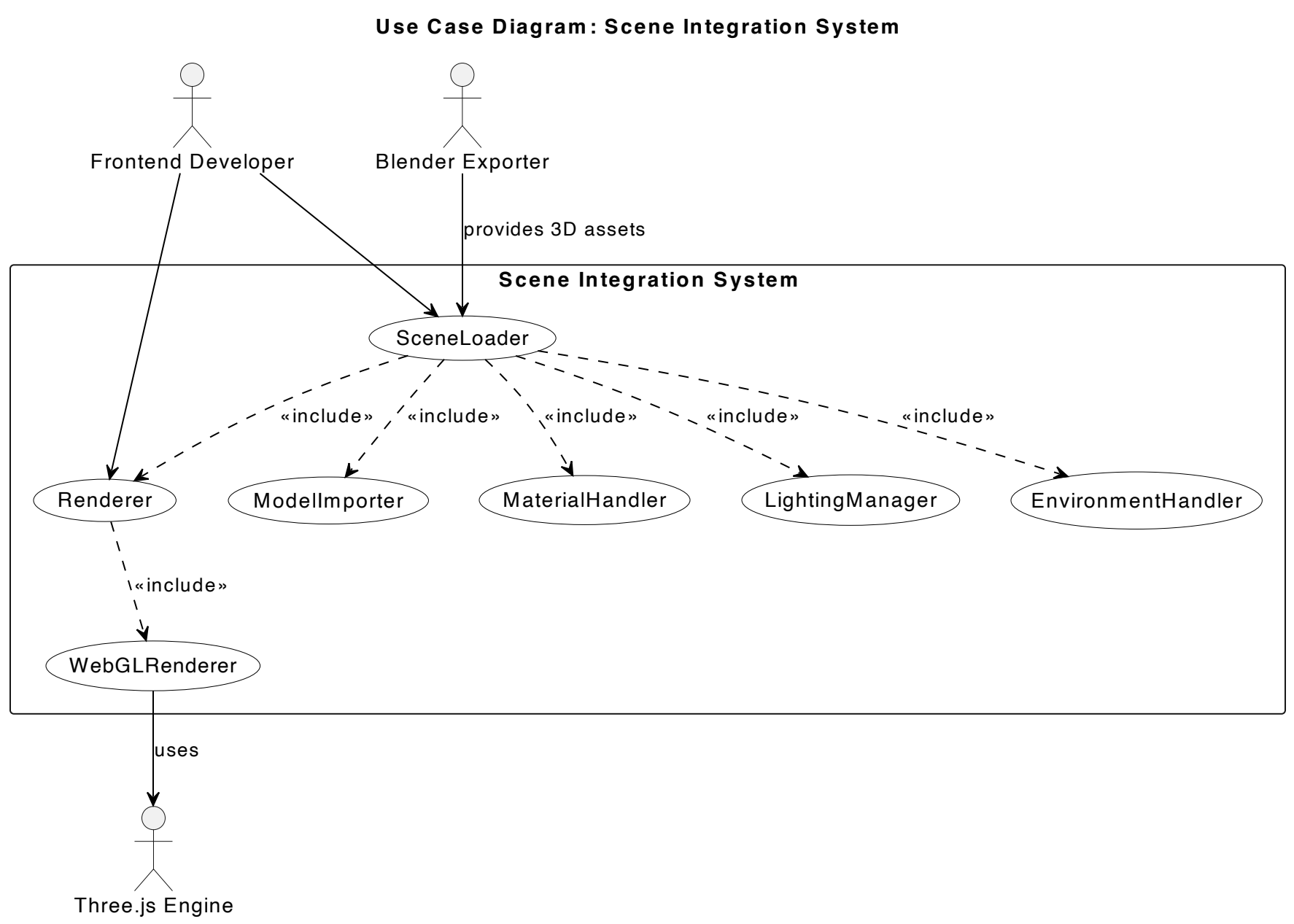


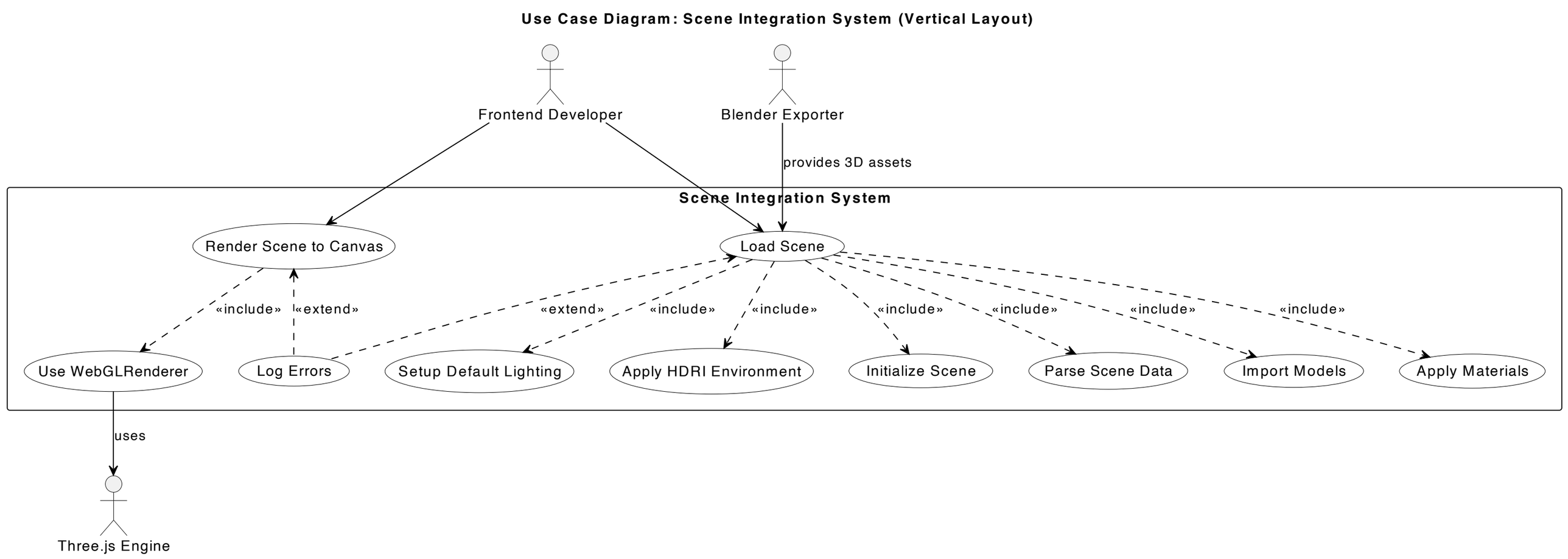
Object Diagram

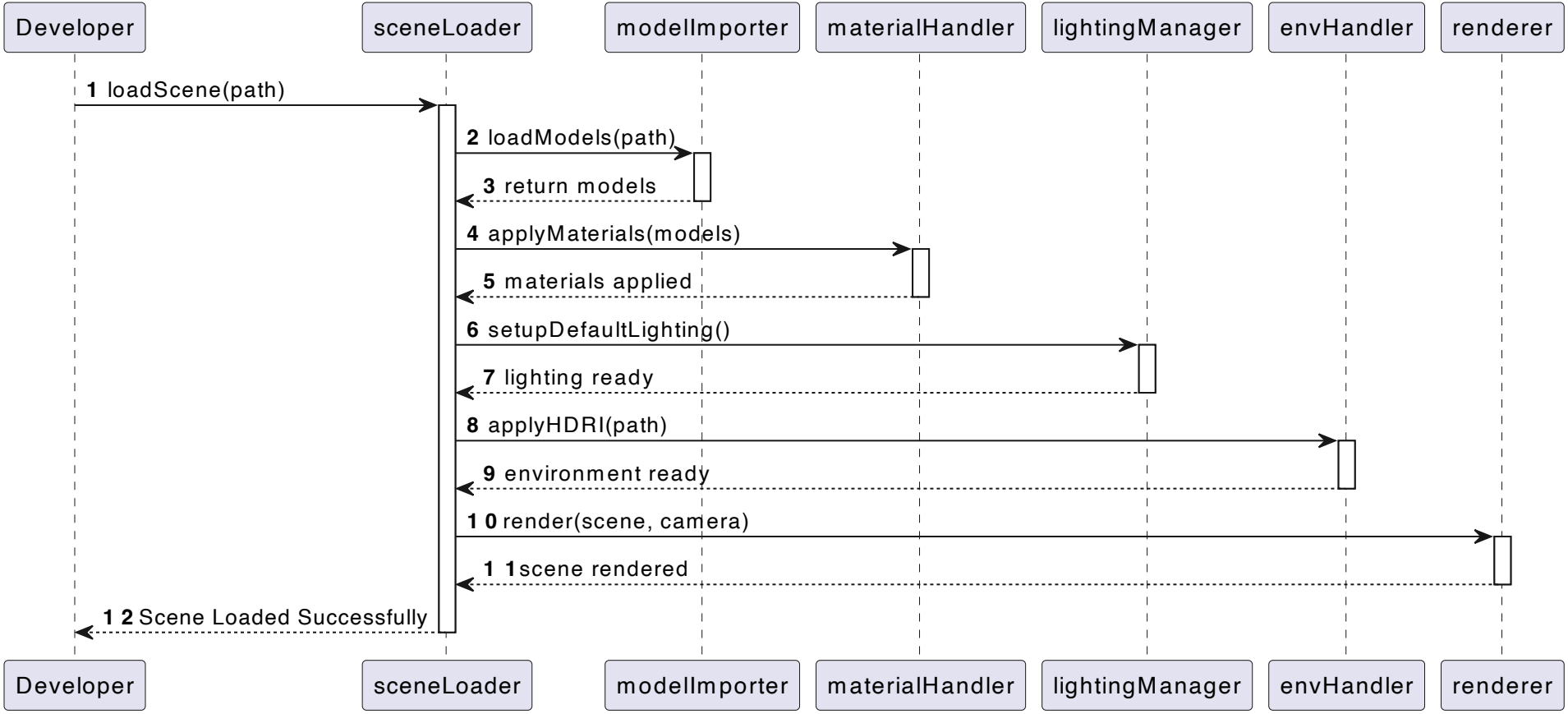


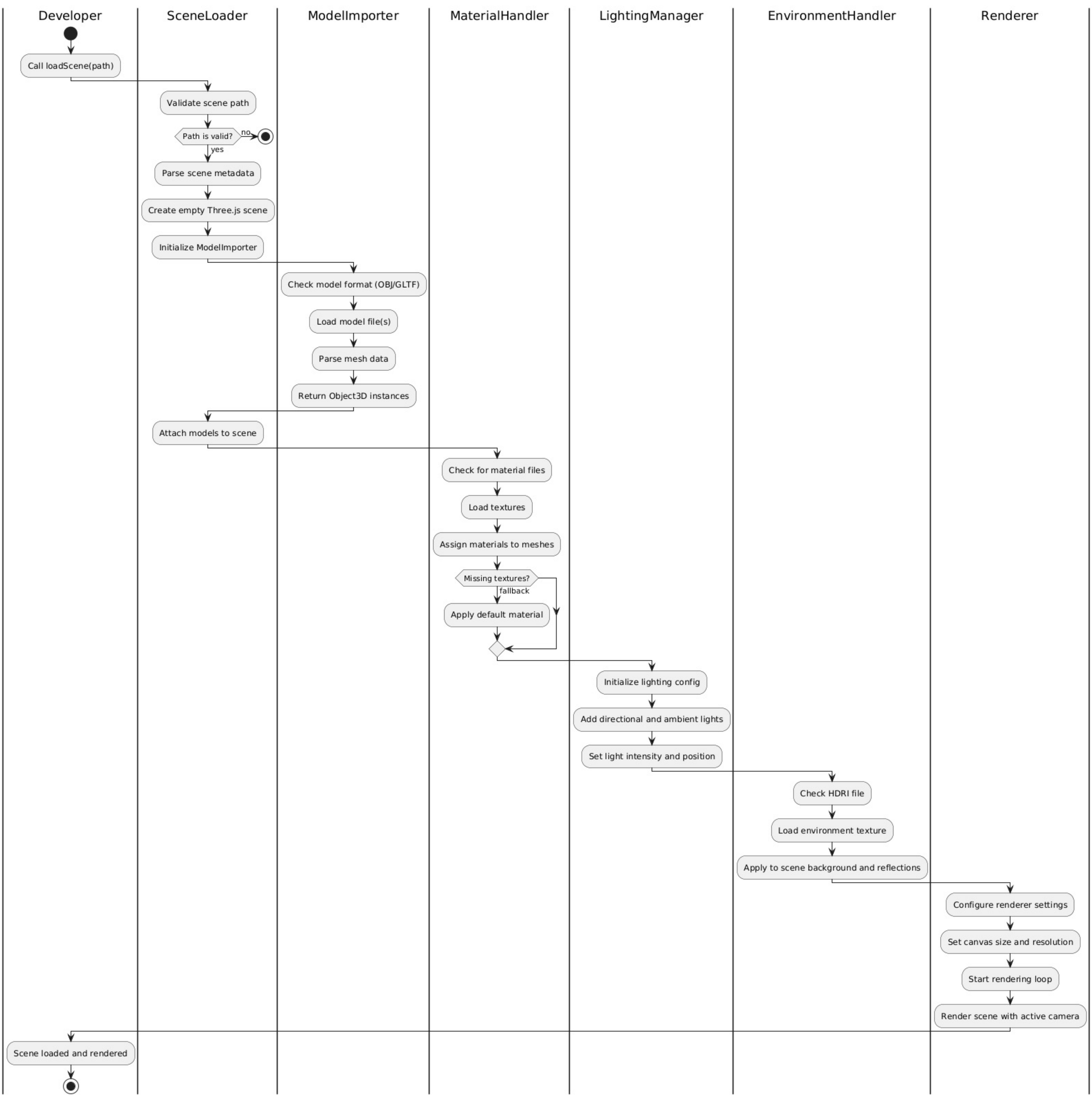
Package Diagram: Scene Integration System

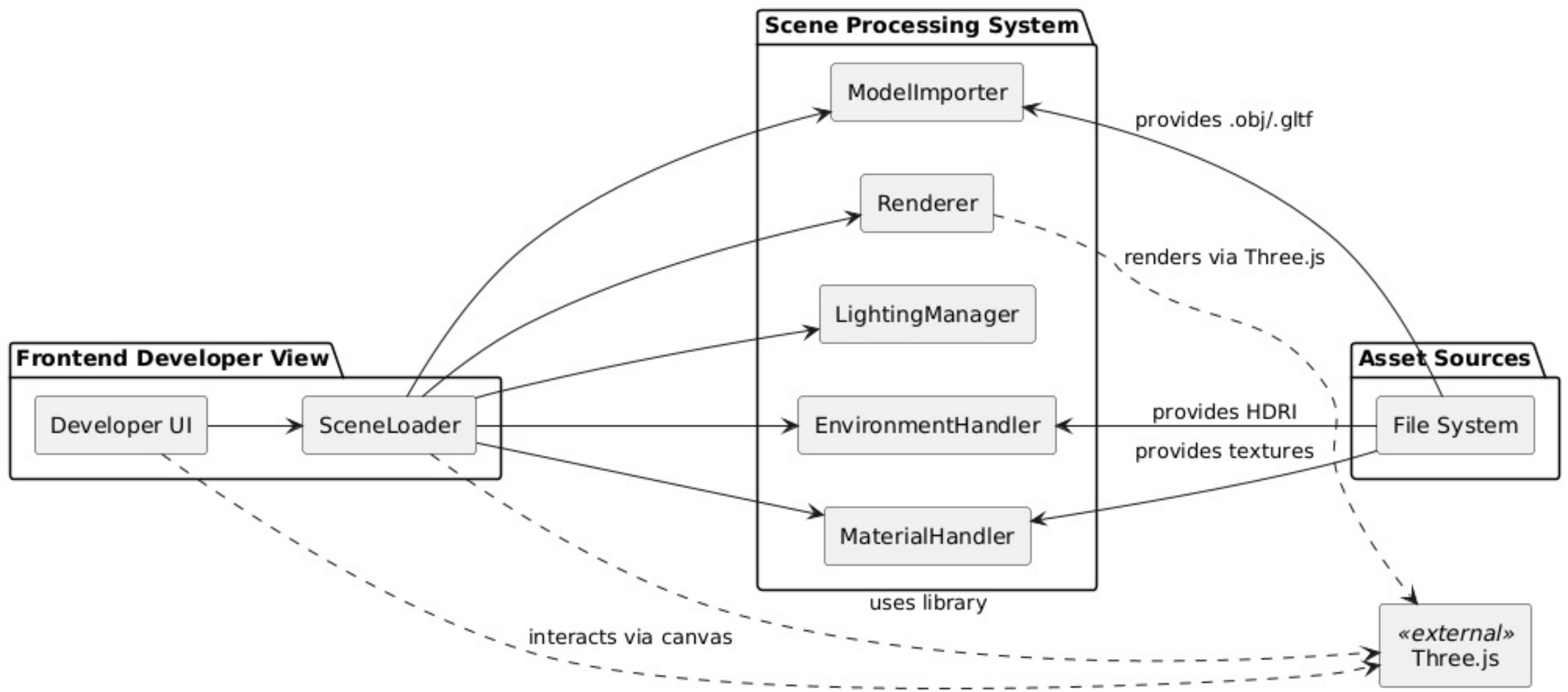


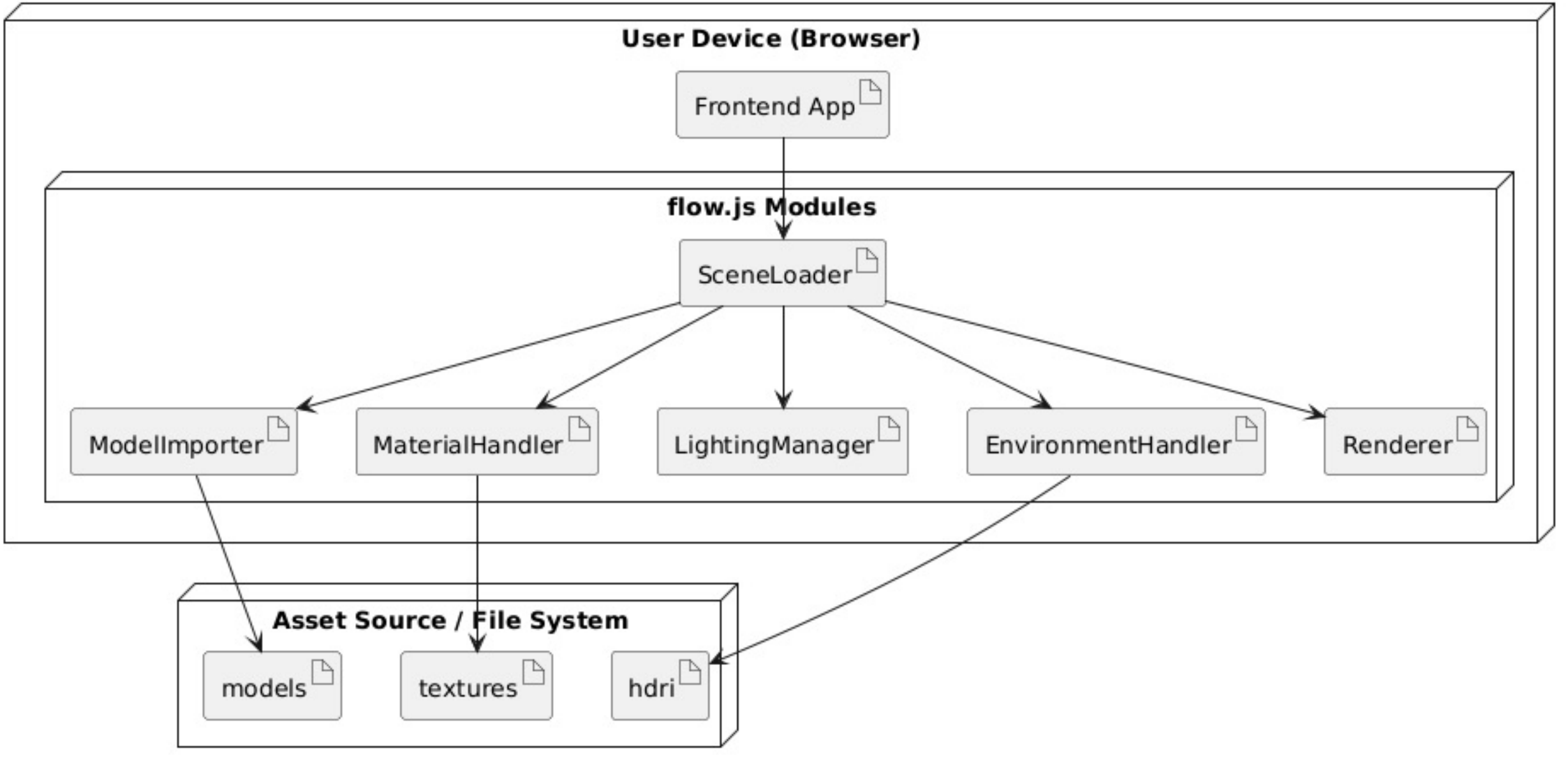

Use Case Diagram: Scene Integration System
Frontend Developer
Blender Exporter
provides 3D assets
Scene Integration System
SceneLoader
«include»
«include»
«include»
«include»
«include»
Renderer
ModelImporter
MaterialHandler
LightingManager
EnvironmentHandler
«include»
WebGLRenderer
uses
Three.js Engine

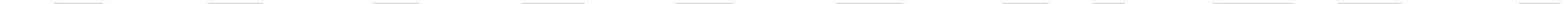

Use Case Diagram: Scene Integration System (Vertical Layout)
Frontend Developer
Blender Exporter
provides 3D assets
Scene Integration System
Render Scene to Canvas
Load Scene
«include»
«extend»
«extend»
«include»
«include»
«include»
«include»
«include»
«include»
Use WebGLRenderer
Log Errors
Setup Default Lighting
Apply HDRI Environment
Initialize Scene
Parse Scene Data
Import Models
Apply Materials
uses
Three.js Engine

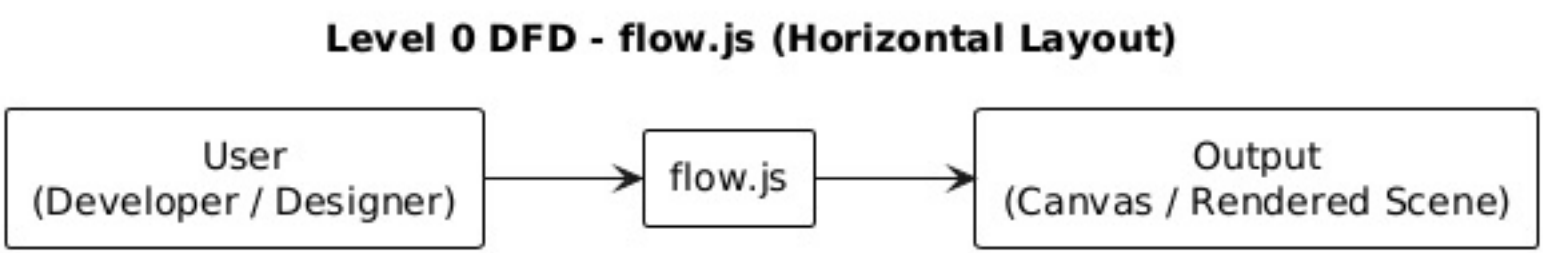

Developer
sceneLoader
modelImporter
materialHandler
lightingManager
envHandler
renderer
1 loadScene(path)
2 loadModels(path)
3 return models
4 applyMaterials(models)
5 materials applied
6 setupDefaultLighting()
7 lighting ready
8 applyHDRI(path)
9 environment ready
10 render(scene, camera)
11 scene rendered
12 Scene Loaded Successfully
Developer
sceneLoader
modelImporter
materialHandler
lightingManager
envHandler
renderer

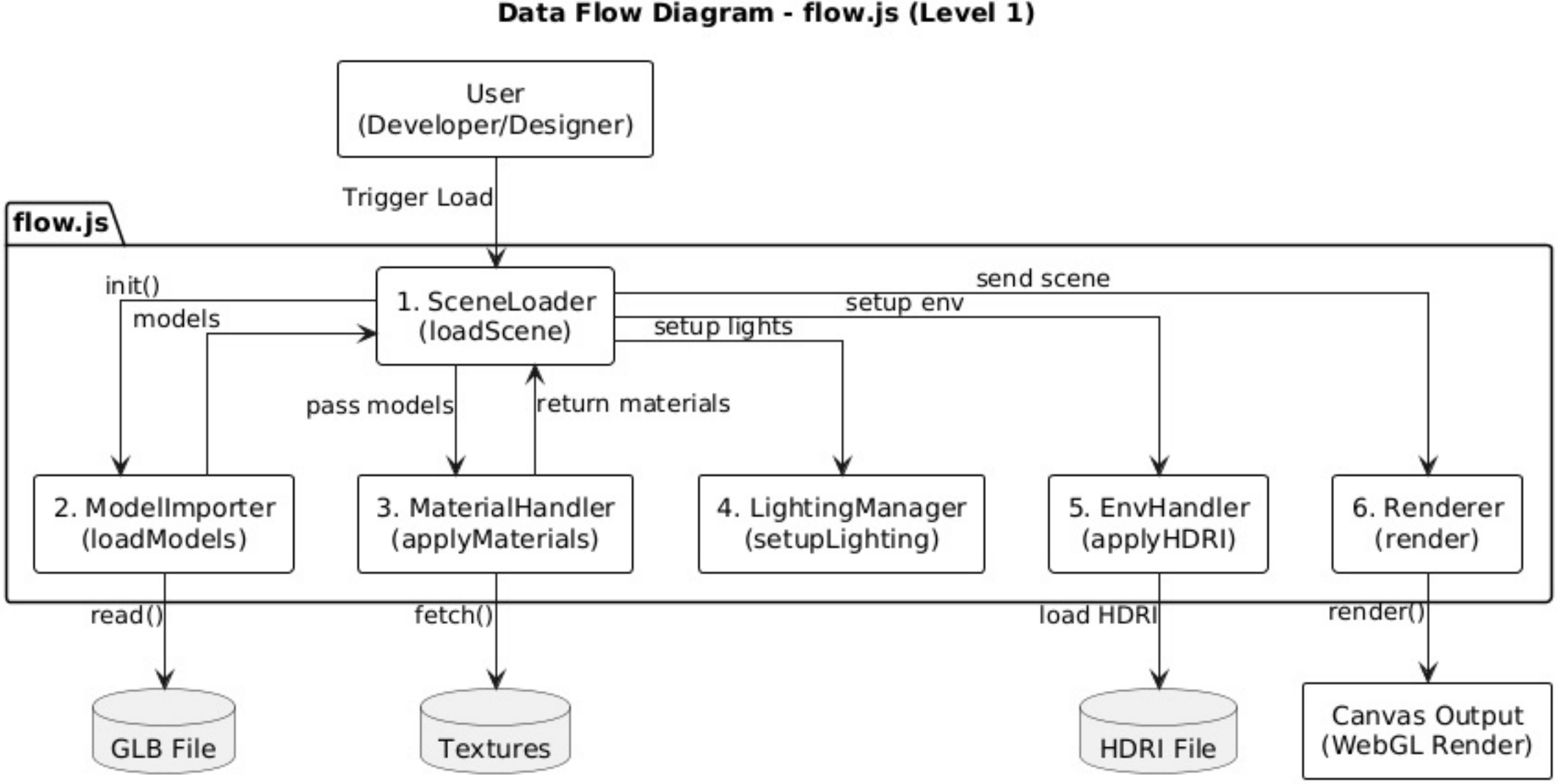

Developer
SceneLoader
ModelImporter
MaterialHandler
LightingManager
EnvironmentHandler
Renderer
Call loadScene(path)
Validate scene path
Path is valid?
no
yes
Parse scene metadata
Create empty Three.js scene
Initialize ModelImporter
Check model format (OBJ/GLTF)
Load model file(s)
Parse mesh data
Return Object3D instances
Attach models to scene
Check for material files
Load textures
Assign materials to meshes
Missing textures?
fallback
Apply default material
Initialize lighting config
Add directional and ambient lights
Set light intensity and position
Check HDRI file
Load environment texture
Apply to scene background and reflections
Configure renderer settings
Set canvas size and resolution
Start rendering loop
Render scene with active camera
Scene loaded and rendered


Scene Processing System
ModelImporter
Renderer
LightingManager
EnvironmentHandler
MaterialHandler
uses library
Frontend Developer View
Developer UI
SceneLoader
Asset Sources
File System
provides .obj/.gltf
renders via Three.js
provides HDRI
provides textures
interacts via canvas
«external»
Three.js


User Device (Browser)
Frontend App
flow.js Modules
SceneLoader
ModelImporter
MaterialHandler
LightingManager
EnvironmentHandler
Renderer
Asset Source / File System
models
textures
hdri


Uninitialized
loadScene(path)
ScenePathSet
parseSceneData()
DataParsed
initializeScene()
SceneInitialized
loadModels()
ModelsLoaded
applyMaterials()
MaterialsApplied
setupLighting()
LightingSet
Dummy
applyHDRI()
EnvironmentApplied
ReadyToRender
render()
Rendering
onFrameComplete()


Level 0 DFD - flow.js (Horizontal Layout)
User
(Developer / Designer)
flow.js
Output
(Canvas / Rendered Scene)


Data Flow Diagram - flow.js (Level 1)
User
(Developer/Designer)
Trigger Load
flow.js
init()
models
1. SceneLoader
(loadScene)
send scene
setup env
setup lights
pass models
return materials
2. ModelImporter
(loadModels)
3. MaterialHandler
(applyMaterials)
4. LightingManager
(setupLighting)
5. EnvHandler
(applyHDRI)
6. Renderer
(render)
read()
fetch()
load HDRI
render()
GLB File
Textures
HDRI File
Canvas Output
(WebGL Render)