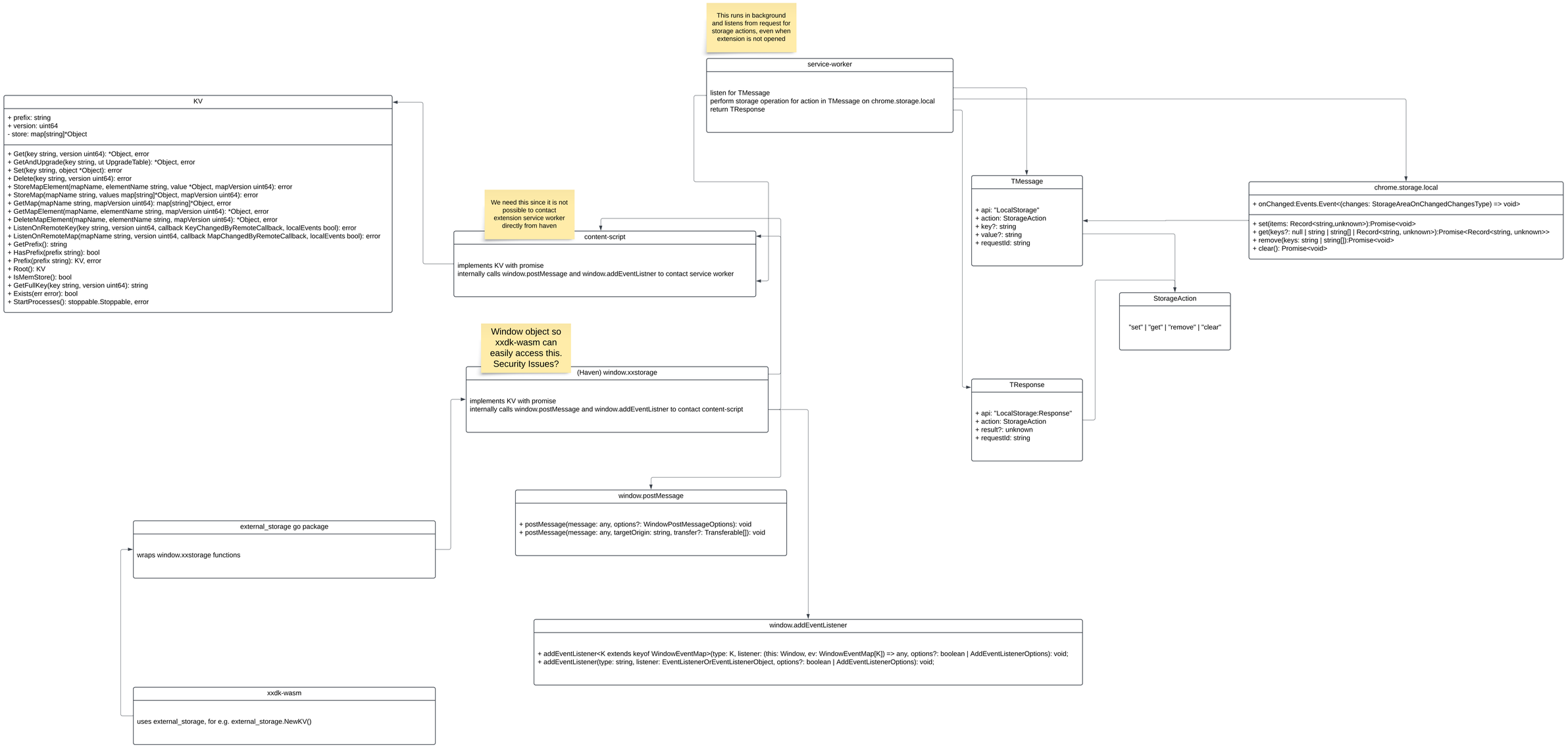
This runs in background and listens from request for storage actions, even when extension is not opened
service-worker
listen for TMessage
perform storage operation for action in TMessage on chrome.storage.local
return TResponse
KV
+ prefix: string
+ version: uint64
- store: map[string]*Object
+ Get(key string, version uint64): *Object, error
+ GetAndUpgrade(key string, ut UpgradeTable): *Object, error
+ Set(key string, object *Object): error
+ Delete(key string, version uint64): error
+ StoreMapElement(mapName, elementName string, value *Object, mapVersion uint64): error
+ StoreMap(mapName string, values map[string]*Object, mapVersion uint64): error
+ GetMap(mapName string, mapVersion uint64): map[string]*Object, error
+ GetMapElement(mapName, elementName string, mapVersion uint64): *Object, error
+ DeleteMapElement(mapName, elementName string, mapVersion uint64): *Object, error
+ ListenOnRemoteKey(key string, version uint64, callback KeyChangedByRemoteCallback, localEvents bool): error
+ ListenOnRemoteMap(mapName string, version uint64, callback MapChangedByRemoteCallback, localEvents bool): error
+ GetPrefix(): string
+ HasPrefix(prefix string): bool
+ Prefix(prefix string): KV, error
+ Root(): KV
+ IsMemStore(): bool
+ GetFullKey(key string, version uint64): string
+ Exists(err error): bool
+ StartProcesses(): stoppable.Stoppable, error
TMessage
+ api: "LocalStorage"
+ action: StorageAction
+ key?: string
+ value?: string
+ requestId: string
chrome.storage.local
+ onChanged:Events.Event<(changes: StorageAreaOnChangedChangesType) => void>
+ set(items: Record<string,unknown>):Promise<void>
+ get(keys?: null | string | string[] | Record<string, unknown>):Promise<Record<string, unknown>>
+ remove(keys: string | string[]):Promise<void>
+ clear(): Promise<void>
We need this since it is not possible to contact extension service worker directly from haven
content-script
implements KV with promise
internally calls window.postMessage and window.addEventListner to contact service worker
StorageAction
"set" | "get" | "remove" | "clear"
Window object so xxdk-wasm can easily access this. Security Issues?
(Haven) window.xxstorage
implements KV with promise
internally calls window.postMessage and window.addEventListner to contact content-script
TResponse
+ api: "LocalStorage:Response"
+ action: StorageAction
+ result?: unknown
+ requestId: string
window.postMessage
+ postMessage(message: any, options?: WindowPostMessageOptions): void
+ postMessage(message: any, targetOrigin: string, transfer?: Transferable[]): void
external_storage go package
wraps window.xxstorage functions
window.addEventListener
+ addEventListener<K extends keyof WindowEventMap>(type: K, listener: (this: Window, ev: WindowEventMap[K]) => any, options?: boolean | AddEventListenerOptions): void;
+ addEventListener(type: string, listener: EventListenerOrEventListenerObject, options?: boolean | AddEventListenerOptions): void;
xxdk-wasm
uses external_storage, for e.g. external_storage.NewKV()