Black Sultan OS – Gesamtes Code-Archiv (ChatGPT Sessions)

Repo Skeleton & HAL

```
#ifndef HAL_H
#define HAL_H
#include <stdint.h>
#include <stddef.h>
#include <stdbool.h>
typedef struct { void *fb_ptr; uint32_t width; uint32_t height; uint32_t stride; uint8_t  bpp; uint32_t pixel_form
typedef enum { HAL_EVT_NONE=0, HAL_EVT_TOUCH, HAL_EVT_KEY } hal_event_type_t;
typedef struct { hal_event_type_t type; uint64_t timestamp; union { struct { int32_t x, y; uint32_t id; uint8_t do
int hal_init(void); void hal_shutdown(void); int hal_get_framebuffer(hal_framebuffer_t *out);
int hal_flush_region(int x, int y, int w, int h); int hal_swap_buffers(void);
int hal_poll_event(hal_event_t *evt); int hal_wait_event(hal_event_t *evt);
uint64_t hal_time_ms(void); void hal_sleep_ms(uint32_t ms);
const char* hal_platform_name(void); int hal_reboot(void);
#endif
```

Demo Framebuffer (C)

```
#include "../hal/include/hal.h"
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/mman.h>
#include <sys/ioctl.h>
#include <linux/fb.h>
#include <unistd.h>
#include <string.h>
#include <time.h>
#include <sys/reboot.h>
int main(int argc, char **argv) {
    if (hal_init() != 0) {
        fprintf(stderr, "Kein /dev/fb0 gefunden\n");
        return 1;
    }
    // Zeichne einfache Animation (Magenta Dreieck)
    ...
}
```

Makefile

```
CC ?= gcc
CFLAGS = -O2 -Wall -Ihal/include
SRCS = src/demo_framebuffer.c
TARGET = bs-demo-fb
all: $(TARGET)
■$(CC) $(CFLAGS) -o $(TARGET) $(SRCS)
clean:
■rm -f $(TARGET) *.o
```

build.sh

```
#!/usr/bin/env bash
set -euo pipefail
echo "Building native demo..."
mkdir -p build
make
echo "Build done. ./bs-demo-fb"
```

docker-compose.yml (Starter)

```
version: "3.8"
services:
  backend:
```

```
    build: ./backend
    ports:
      - "8000:8000"
    volumes:
      - ./backend:/app
  frontend:
    image: nginx:stable-alpine
    volumes:
      - ./frontend:/usr/share/nginx/html:ro
    ports:
      - "3000:80"
    depends_on:
      - backend
```

backend/main.py (FastAPI)

```
from fastapi import FastAPI, WebSocket, WebSocketDisconnect
import asyncio, json, uvicorn, time
app = FastAPI(title="Black Sultan OS - Safe Demo Backend")
STATE = {"nodes": [], "connections": [], "intelligence":0, "threats":[], "events":[]}
class EventBus: ...
bus = EventBus()
@app.get("/api/state")
async def get_state(): return STATE
@app.websocket("/ws")
async def ws(ws: WebSocket):
    await bus.register(ws)
    await ws.send_text(json.dumps({"type":"init","state":STATE}))
    ...
if __name__ == "__main__":
    uvicorn.run("main:app", host="0.0.0.0", port=8000)
```

frontend/index.html (UI Auszug)

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Black Sultan Starter UI</title>
  <style>
    body { background:#0b0b0d; color:#e6eef6; font-family: monospace; }
    .map { position:relative; height:400px; background:#06121a; }
    .node { position:absolute; width:10px; height:10px; border-radius:50%; }
  </style>
</head>
<body>
  <h1>Black Sultan Starter</h1>
  <div id="worldMap" class="map"></div>
  <input id="commandInput" type="text">
  <script>
    const ws = new WebSocket("ws://"+location.hostname+":8000/ws");
    ws.onmessage = ev => console.log(ev.data);
  </script>
</body>
</html>
```