

Micro Dao Orchestrator Ui — React Layout (shell)· typescript

```
import React from "react";
<Input placeholder="Пошук по всіх модулях" className="max-w-md" />
<Tabs defaultValue="dao" className="ml-2">
  <TabsList>
    <TabsTrigger value="private">Private</TabsTrigger>
    <TabsTrigger value="dao">DAO</TabsTrigger>
    <TabsTrigger value="public">Public</TabsTrigger>
  </TabsList>
</Tabs>
<div className="ml-auto flex items-center gap-2">
  <Badge variant="outline" className="flex items-center gap-1">
    <Zap className={`h-4 w-4 ${netOnline ? 'text-emerald-600' : 'text-amber-600'}` />
    {netOnline ? 'Online' : 'Offline'}
  </Badge>
  <Badge variant="outline" className="flex items-center gap-1">
    <Database className={`h-4 w-4 ${orchOk ? 'text-emerald-600' : 'text-amber-600'}` />
    Orchestrator {orchOk ? 'ok' : 'unreachable'}
  </Badge>
  <Button variant="secondary" size="sm" className="gap-2"><Settings className="h-4 w-4" />Налаштування</Button>
</div>
</div>
);
}
```

```
function HealthGrid() {
  const items = [
    { title: "Messenger", ok: true },
    { title: "Parser", ok: false },
    { title: "KB Core", ok: true },
    { title: "RAG", ok: true },
    { title: "Wallet", ok: true },
    { title: "DAO", ok: true },
  ];
  return (
    <div className="grid grid-cols-2 lg:grid-cols-3 gap-3">
      {items.map((x) => (
        <Card key={x.title} className="rounded-2xl">
          <CardHeader className="py-3">
            <CardTitle className="text-sm font-medium flex items-center gap-2">
              {x.ok ? (
                <CheckCircle2 className="h-4 w-4 text-emerald-600" />
              ) : (
                <AlertTriangle className="h-4 w-4 text-amber-600" />
              )}
              {x.title}
            </CardTitle>
          </CardHeader>
        </Card>
      )}
    </div>
  );
}
```

```

</CardHeader>
<CardContent className="py-2">
<div className="text-xs text-slate-500">{x.ok ? "Працює стабільно" : "Черга задач >
р95"}</div>
</CardContent>
</Card>
)}}
</div>
);
}

function OrchestratorChat() {
return (
<Card className="rounded-2xl h-full flex flex-col">
<CardHeader className="py-3">
<CardTitle className="text-sm">Чат з Оркестратором</CardTitle>
</CardHeader>
<CardContent className="flex-1 flex flex-col gap-3">
<div className="flex-1 rounded-xl border bg-slate-50 p-3 text-sm overflow-auto">
<div className="text-slate-500">Вітаю. Чим допомогти? Наприклад: "Розбери PDF та
створй короткий бріф у Проєктах".</div>
</div>
<div className="flex items-center gap-2">
<Textarea placeholder="Запитайте або використайте @agent" className="min-h-[44px]"
/>
<Button className="whitespace-nowrap">Надіслати</Button>
</div>
</CardContent>
</Card>
);
}

function ActivityFeed() {
const rows = [
{ t: "ingest.completed", d: "USDO готово", ts: "09:15" },
{ t: "message.created", d: "#general", ts: "09:12" },
{ t: "vote.finalized", d: "Постанова #12", ts: "08:55" },
{ t: "payment.sent", d: "Reward 2.5 μUTIL", ts: "08:40" },
];
return (
<Card className="rounded-2xl h-full">
<CardHeader className="py-3"><CardTitle className="text-sm">Стрічка
подій</CardTitle></CardHeader>
<CardContent>
<ul className="space-y-2 text-sm">
{rows.map((r, i) => (
<li key={i} className="flex items-center gap-2">
<Activity className="h-4 w-4 text-slate-500"/>

```

```
<span className="font-medium">{r.t}</span>
<span className="text-slate-500">— {r.d}</span>
<span className="ml-auto text-xs text-slate-400">{r.ts}</span>
</li>
)}}}
```

Зміни потрібні. Я спробував застосувати правки в Canvas автоматично, але інструмент не знайшов кілька точних збігів у вашій версії файлу і відхилив патч.

Щоб не блокуватися, ось мінімальні дифи для ручної правки у вашому поточному файлі **“Micro Dao Orchestrator Ui — React Layout (shell)”**. Вони точно відповідають його актуальному вмісту вище.

## 1) Торбар: 3-статус оркестратора

### A. Сигнатура:

```
- function Topbar({ netOnline, orchOk }: { netOnline: boolean;
orchOk: boolean }) {
+ function Topbar({ netOnline, orchStatus }: { netOnline: boolean;
orchStatus: 'ok' | 'down' | 'unknown' }) {
```

### B. Бейдж Orchestrator:

```
- <Database className={ `h-4 w-4 ${orchOk ? 'text-emerald-600' :
'text-amber-600'} ` } />
- Orchestrator {orchOk ? 'ok' : 'unreachable'}
+ <Database className={ `h-4 w-4 ${orchStatus === 'ok' ?
'text-emerald-600' : 'text-amber-600'} ` } />
+ Orchestrator {orchStatus === 'ok' ? 'ok' : orchStatus === 'down' ?
'down' : 'unknown (CORS?)' }
```

### C. Виклик Topbar у return:

```
- <Topbar netOnline={netOnline} orchOk={orchOk} />
+ <Topbar netOnline={netOnline} orchStatus={orchStatus} />
```

### D. Стейт у OrchestratorLayout():

```
- const [orchOk, setOrchOk] = React.useState<boolean>(true);
+ const [orchStatus, setOrchStatus] =
React.useState<'ok' | 'down' | 'unknown'>('unknown');
```

## 2) Блок helpers: /healthz з конфігом, рекомендація профілю, оцінка індексу

Замініть усе від рядка `// --- helpers ---` до початку `export default function OrchestratorLayout()` на це:

```
// --- helpers ---
const CONFIG = {
  HEALTHZ_URL: (typeof import !== 'undefined' && (import.meta as any)?.env?.VITE_HEALTHZ_URL) || '/healthz',
};

async function pingHealthz(url: string, timeoutMs = 3000):
Promise<'ok'|'down'|'unknown'> {
  try {
    const ctrl = new AbortController();
    const t = setTimeout(() => ctrl.abort(), timeoutMs);
    const res = await fetch(url, { signal: ctrl.signal, headers: {
accept: 'application/json,text/plain,*/*' } });
    clearTimeout(t);
    return res.ok ? 'ok' : 'down';
  } catch {
    const online = typeof navigator !== 'undefined' ?
navigator.onLine : false;
    return online ? 'unknown' : 'down';
  }
}
```

```
async function recommendModelProfile():
Promise<'Lite'|'Base'|'Plus'|'Pro'> {
  // Heuristics: deviceMemory (GB), CPU cores, simple UA.
  // @ts-ignore
  const dm = (navigator as any).deviceMemory || 4;
  const cores = navigator.hardwareConcurrency || 4;
  const isMobile = /iPhone|Android/i.test(navigator.userAgent);
  if (dm >= 24 && cores >= 8 && !isMobile) return 'Pro';
  if (dm >= 12 && cores >= 8) return 'Plus';
  if (dm >= 6) return 'Base';
}
```

```

    return 'Lite';
}

function profileSizeMB(p: 'Lite'|'Base'|'Plus'|'Pro'): number {
    return p === 'Lite' ? 300 : p === 'Base' ? 1000 : p === 'Plus' ?
4000 : 7000;
}

function parseIndexSizeMB(label: string): number | null {
    if (!label) return null;
    if (label.startsWith('custom:')) return null;
    if (label.toUpperCase().endsWith('GB')) return parseInt(label) *
1024;
    if (label.toUpperCase().endsWith('MB')) return parseInt(label);
    return null;
}

// ~2KB/chunk → ≈512 чанків/MB
function estimateChunksInt8(sizeMB: number): number { return
Math.floor(sizeMB * 512); }

```

### 3) OrchestratorLayout(): стейти, useEffect, інсталятор, пропси StartScreen

#### А. Додайте/оновіть стейти:

```

const [modelProfile, setModelProfile] =
React.useState<string>('Base');
const [recommendedProfile, setRecommendedProfile] =
React.useState<string>('Base');
const [indexSize, setIndexSize] = React.useState<string>('500MB');
+ const [estChunks, setEstChunks] = React.useState<string>('');
const [netOnline, setNetOnline] = React.useState<boolean>(typeof
navigator !== 'undefined' ? navigator.onLine : true);
- const [orchOk, setOrchOk] = React.useState<boolean>(true);
+ const [orchStatus, setOrchStatus] =
React.useState<'ok'|'down'|'unknown'>('unknown');

```

## **В. Замініть увесь `useEffect` на:**

```
React.useEffect(() => {
  const on = () => setNetOnline(true);
  const off = () => setNetOnline(false);
  window.addEventListener('online', on);
  window.addEventListener('offline', off);

  let mounted = true;

  // real /healthz ping every 10s with config + CORS aware
  const tick = async () => {
    const st = await pingHealthz(CONFIG.HEALTHZ_URL);
    if (mounted) setOrchStatus(st);
  };
  tick();
  const id = setInterval(tick, 10000);

  // model profile recommendation
  recommendModelProfile().then(p => { if (mounted) {
setRecommendedProfile(p); setModelProfile(p); } });

  // restore custom index path if saved
  const savedPath = localStorage.getItem('microdao.indexPath');
  if (savedPath) setIndexSize(`custom:${savedPath}`);

  // estimate chunks for selected index size
  const estId = setInterval(() => {
    const mb = parseIndexSizeMB(indexSize);
    if (mb) setEstChunks(`≈
${Math.round(estimateChunksInt8(mb)/1000)} тис.`);
    else setEstChunks('');
  }, 300);

  return () => { mounted = false;
window.removeEventListener('online', on);
window.removeEventListener('offline', off); clearInterval(id);
clearInterval(estId); };
}, [indexSize]);
```

### C. Інстальатор ваг, прив'язаний до профілю/індексу:

```
const handleInstallAgent = () => {
  if (installing || agentReady) return;
  setInstalling(true);
  setProgress(0);
-   const timer = setInterval(() => {
-     setProgress((p) => {
-       const next = Math.min(100, p + Math.random() * 18 + 5);
-       if (next >= 100) {
-         clearInterval(timer);
-         setInstalling(false);
-         setAgentReady(true);
-       }
-       return next;
-     });
-   }, 400);
+   localStorage.setItem('microdao.modelProfile', modelProfile);
+   localStorage.setItem('microdao.indexSize', indexSize);
+   const modelMB = profileSizeMB(modelProfile as any);
+   const idxMB = parseIndexSizeMB(indexSize) || 0;
+   const totalMB = modelMB + idxMB;
+   const estTimeMs = Math.min(120000, 1500 * (totalMB / 50)); //
~1.5s per 50MB, cap 2min
+   const started = Date.now();
+   const timer = setInterval(() => {
+     const elapsed = Date.now() - started;
+     let pct = Math.min(100, (elapsed / estTimeMs) * 100);
+     pct = pct < 100 ? (pct * 0.85) : 100; // ease-out
+     setProgress(pct);
+     if (pct >= 99.5) {
+       clearInterval(timer);
+       setInstalling(false);
+       setAgentReady(true);
+     }
+   }, 300);
  };
};
```

### D. Пропси для StartScreen у content ( ) (внизу):

```
<StartScreen
  ...
```

```
    recommendedProfile={recommendedProfile}
-   onPickCustomIndex={handlePickCustomIndex}
+   onPickCustomIndex={handlePickCustomIndex}
+   estChunks={estChunks}
  />
```

## 4) StartScreen: додайте проп і підказку

### А. Сигнатура:

```
- function StartScreen({ onCreateDAO, onJoinDAO, onSolo, agentReady,
onInstallAgent, installing, progress, modelProfile, setModelProfile,
indexSize, setIndexSize, recommendedProfile, onPickCustomIndex }):
+ function StartScreen({ onCreateDAO, onJoinDAO, onSolo, agentReady,
onInstallAgent, installing, progress, modelProfile, setModelProfile,
indexSize, setIndexSize, recommendedProfile, onPickCustomIndex,
estChunks }):
```

...та в типі параметрів додайте `estChunks: string;`

### В. Підказка під індексом:

```
- <div className="text-xs text-slate-500">int8 квантування заощаджує
×3-4 місця.</div>
+ <div className="text-xs text-slate-500">
+   int8 квантування заощаджує ×3-4 місця. {estChunks && <span
className="ml-1">Орієнтовно: {estChunks} чанків (int8).</span>}
+ </div>
```

Після цих замін у вас буде:

- `/healthz` з URL із конфіга і CORS-статусом `unknown`.
- Автовибір профілю моделі та збереження вибору.
- Оцінка місткості індексу в чанках (int8).
- Тристатусний бейдж оркестратора.

