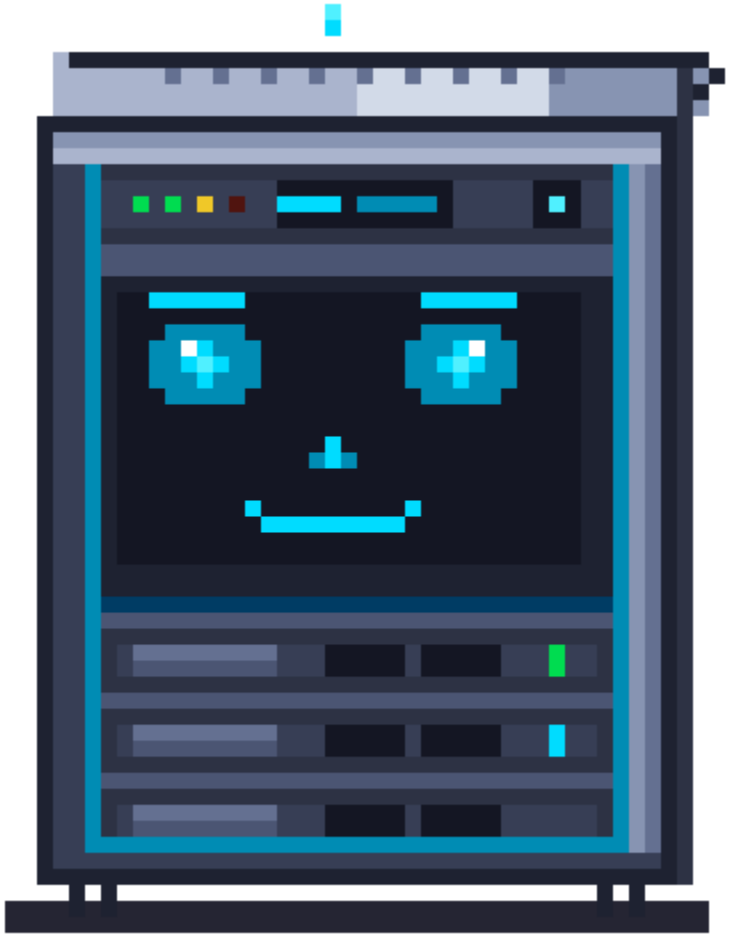
?



HELIKON

SEND

LLM PROVIDER

Local | Claude | ↔ Dual

Runden: ? 6 ? (3-20) Zusammenfassung

TOOLS

? Stats | PDF | ? Vollbild

```
(function() {

    /* ============================================================
     * HELIKON BUDDY CHAT v2.6.8 — 2026-09-25
     * --- firnCore customization ---
     * Doku-Suche auf /site/query (bestguess) umgestellt, Prompt-
     * Bereinigung, isShortQuery-Gate entfernt, Kontext-Ausschnitt
```

```
     * rund um den Treffer, Datei-Treffer im Kontext.
     * ========================================================== */

    var fontLink = document.createElement('link');
    fontLink.rel = 'stylesheet';
    fontLink.href = 'https://fonts.googleapis.com/
css2?family=Press+Start+2P';
    document.head.appendChild(fontLink);

    var PF = '"Press Start 2P","Courier New",Courier,monospace';
    var CRISP = '-webkit-font-smoothing:none;
-moz-osx-font-smoothing:unset; font-smooth:never;
text-rendering:optimizeSpeed;';

    var css = ''
    + '#serverBuddy { display: none !important; }'
    + '#buddySpeech { display: none !important; }'
    + '#buddyFrame { margin:0; border:2px solid #ccc; border-radius:8px;
background:#f8f8f8; padding:14px; font-family:' + PF + '; color:#333; ' +
CRISP + ' }'
    + '#buddyChat { display:flex; gap:14px; align-items:flex-start; }'
    + '#buddySide { width:200px; flex-shrink:0; display:flex;
flex-direction:column; align-items:center; border:2px solid #ddd;
border-radius:6px; background:#f0f0f0; padding:12px 8px; position:sticky;
top:20px; user-select:none; -webkit-user-select:none; }'
    + '#buddySide.dragging { position:fixed; z-index:9999;
pointer-events:auto; }'
    + '#buddyDragHandle { font-size:12px; color:#ccc; cursor:grab;
padding:2px 8px; border-radius:4px; transition:color 0.2s ease;
align-self:flex-end; margin-top:-4px; }'
    + '#buddyDragHandle:hover { color:#0077aa; }'
    + '#buddySide.dragging #buddyDragHandle { color:#0077aa;
cursor:grabbing; }'
    + '#buddyChatImg { width:160px; height:160px;
image-rendering:pixelated; image-rendering:crisp-edges; display:block;
border:none; cursor:pointer; transition:transform 0.15s ease; }'
    + '#buddyChatImg:hover { transform:rotate(-3deg) scale(1.05); }'
    + '#buddyNameTag { font-size:7px; color:#999; letter-spacing:3px;
text-transform:uppercase; margin-top:6px; }'
    + '#buddyBubble { transform:scale(0); background:#fff; color:#333;
padding:10px 14px; border-radius:12px; border:2px solid #ccc;
font-family:' + PF + '; font-size:9px; line-height:1.8; max-width:180px;
text-align:center; white-space:pre-wrap; word-wrap:break-word; opacity:0;
transition:opacity 0.3s ease,transform 0.3s ease; box-shadow:0 2px 8px
rgba(0,0,0,0.08); margin-bottom:8px; position:relative; ' + CRISP + ' }'
    + '#buddyBubble.visible { opacity:1; transform:scale(1); }'
    + '#buddyBubble::after { content:""; position:absolute; bottom:-8px;
left:50%; transform:translateX(-50%); width:0; height:0; border-left:8px
solid transparent; border-right:8px solid transparent; border-top:8px
solid #ccc; }'
```

```
    + '#buddyMain { flex:1; min-width:0; }'
    + '#chatLog { min-height:350px; max-height:550px; overflow-y:auto;
padding:16px; border:2px solid #ddd; border-radius:6px 6px 0 0;
background:#fff; }'
    + '.chatMsg { margin-bottom:10px; padding:10px 14px;
border-radius:6px; font-size:12px; line-height:1.8; max-width:90%;
word-wrap:break-word; }'
    + '.chatMsg.user { background:#e8e8e8; color:#555; margin-left:auto;
text-align:right; border:1px solid #ddd; }'
    + '.chatMsg.buddy { background:#f8f8f8; color:#333;
margin-right:auto; border:1px solid #ddd; }'
    + '.chatMsg.buddy::before { content:"> "; color:#0077aa;
margin-right:4px; opacity:0.6; }'
    + '.chatMsg.buddy.claude { border-left:3px solid #d97706; }'
    + '.chatMsg.buddy.local { border-left:3px solid #3a8f3a; }'
    + '.chatMsg.buddy .msgProvider { display:inline-block; font-size:7px;
padding:2px 6px; border-radius:3px; margin-bottom:4px;
letter-spacing:0.5px; text-transform:uppercase; }'
    + '.chatMsg.buddy .msgProvider.prov-local { background:#e8ffe8;
color:#3a8f3a; border:1px solid #3a8f3a; }'
    + '.chatMsg.buddy .msgProvider.prov-claude { background:#fff8e8;
color:#d97706; border:1px solid #d97706; }'
    + '.chatMsg.system { color:#999; font-size:10px; text-align:center;
margin:6px auto; max-width:100%; }'
    + '.chatMsg.typing { color:#999; opacity:0.6; }'
    + '.chatMsg.fact { color:#0077aa; font-size:10px; text-align:left;
margin:6px 0; max-width:100%; border-left:3px solid #0077aa; padding:8px
12px; background:#f0f8ff; opacity:0.9; }'
    + '.chatMsg.fact::before { content:"// "; opacity:0.5; }'
    + '.chatMsg.dual-chain { color:#d97706; font-size:9px;
text-align:center; margin:4px auto; max-width:100%; opacity:0.7; }'
    + '.chatMsg a { color:#0077aa; text-decoration:underline; }'
    + '.chatMsg a:hover { color:#005580; }'
    + '.chatMsg pre { background:#1a1a2e !important; color:#e8e8f0
!important; padding:14px 16px !important; border-radius:6px !important;
overflow-x:auto !important; font-family:"Courier New",monospace
!important; font-size:13px !important; line-height:1.8 !important;
margin:10px 0 !important; white-space:pre-wrap !important;
word-wrap:break-word !important; border:2px solid #4a4a7a !important;
box-shadow:inset 0 2px 6px rgba(0,0,0,0.4) !important; }'
    + '.chatMsg pre * { color:#e8e8f0 !important; }'
    + '.chatMsg pre .cm { color:#6a9955 !important; font-style:italic; }'
    + '.chatMsg pre .kw { color:#569cd6 !important; }'
    + '.chatMsg pre .st { color:#ce9178 !important; }'
    + '.chatMsg pre .nm { color:#9cdcfe !important; }'
    + '.chatMsg code { background:#2a2a3e !important; color:#f0a868
!important; padding:3px 8px !important; border-radius:3px !important;
font-family:"Courier New",monospace !important; font-size:12px
!important; border:1px solid #5a5a8a !important; }'
    + '.chatMsg pre code { background:transparent !important;
```

```
color:#e8e8f0 !important; padding:0 !important; border:none !important;
font-size:13px !important; }'
    + '.chatMsg ul, .chatMsg ol { margin:6px 0 6px 20px; padding:0; }'
    + '.chatMsg li { margin:4px 0; }'
    + '.chatMsg .mdH { font-size:12px; color:#0077aa; margin:10px 0 4px
0; display:block; }'
    + '#chatInputWrap { display:flex; gap:0; border:2px solid #ddd;
border-top:none; border-radius:0 0 6px 6px; overflow:hidden; }'
    + '#chatInput { flex:1; background:#fff; border:none; color:#333;
font-family:' + PF + '; font-size:12px; padding:16px 14px; outline:none;
' + CRISP + ' }'
    + '#chatInput::placeholder { color:#bbb; font-size:8px; }'
    + '#chatInput:focus { background:#f0f8ff; }'
    + '#chatSend { background:#3a8f3a; color:#fff; border:none;
padding:18px 36px; font-family:' + PF + '; font-size:16px;
cursor:pointer; letter-spacing:3px; transition:all 0.2s ease;
min-width:130px; ' + CRISP + ' }'
    + '#chatSend:hover { background:#2a6f2a; }'
    + '#chatSend:active { transform:scale(0.97); }'
    + '#chatSend:disabled { background:#ccc; color:#999; cursor:default;
transform:none; }'
    + '.buddyModule { border:2px solid #ddd; border-radius:6px;
padding:10px 12px; margin-top:10px; background:#f0f0f0; }'
    + '.modTitle { font-size:7px; color:#0077aa; letter-spacing:2px;
text-transform:uppercase; margin:0 0 8px 0; padding:0 0 6px 0;
border-bottom:1px solid #ddd; }'
    + '#llmSwitch { display:flex; gap:6px; align-items:center;
flex-wrap:wrap; }'
    + '.llmBtn { background:#fff; border:2px solid #ccc; color:#666;
padding:8px 14px; font-family:' + PF + '; font-size:9px; cursor:pointer;
border-radius:4px; transition:all 0.2s ease; display:inline-flex;
align-items:center; gap:6px; ' + CRISP + ' }'
    + '.llmBtn:hover { border-color:#0077aa; color:#0077aa; }'
    + '.llmBtn.active { background:#e8f4ff; color:#0077aa;
border-color:#0077aa; }'
    + '.llmBtn.active-dual { background:#fff8e8; color:#d97706;
border-color:#d97706; }'
    + '.llmBtn:disabled { opacity:0.3; cursor:default; }'
    + '.llmBtn:disabled:hover { border-color:#ccc; color:#666; }'
    + '.llmLed { display:inline-block; width:8px; height:8px; border:1px
solid #aaa; background:#ccc; flex-shrink:0; transition:all 0.3s ease; }'
    + '.llmLed.ok { background:#3a8f3a; border-color:#2a6f2a;
box-shadow:0 0 4px rgba(58,143,58,0.4); }'
    + '.llmLed.warn { background:#cc8800; border-color:#aa6600;
box-shadow:0 0 4px rgba(204,136,0,0.4); }'
    + '.llmLed.err { background:#cc3333; border-color:#aa2222;
box-shadow:0 0 6px rgba(204,51,51,0.5); }'
    + '#dualInfo { display:none; margin-top:8px; padding:8px 10px;
border:2px dashed #d97706; border-radius:6px; background:#fffbf0;
font-size:8px; color:#996600; line-height:1.8; }'
```

```
    + '#dualInfo.visible { display:block; }'
    + '#dualInfoText { margin-bottom:6px; }'
    + '#dualRounds { display:flex; align-items:center; gap:6px;
padding:6px 0 0 0; border-top:1px dashed #e8c878; flex-wrap:wrap; }'
    + '#dualRoundsLabel { font-size:7px; color:#996600;
letter-spacing:1px; text-transform:uppercase; }'
    + '#dualRndDown, #dualRndUp { background:#fff; border:1px solid
#d97706; color:#d97706; width:22px; height:22px; font-size:8px;
cursor:pointer; border-radius:3px; transition:all 0.15s ease;
display:flex; align-items:center; justify-content:center; padding:0;
font-family:' + PF + '; }'
    + '#dualRndDown:hover, #dualRndUp:hover { background:#d97706;
color:#fff; }'
    + '#dualRndVal { font-size:12px; color:#d97706; min-width:22px;
text-align:center; font-weight:bold; }'
    + '#dualRndRange { font-size:6px; color:#bb8833; }'
    + '#dualSummaryWrap { display:inline-flex; align-items:center;
gap:5px; margin-left:10px; padding-left:10px; border-left:1px dashed
#e8c878; cursor:pointer; user-select:none; -webkit-user-select:none; }'
    + '#dualSummaryCheck { appearance:none; -webkit-appearance:none;
width:12px; height:12px; border:1px solid #d97706; border-radius:2px;
background:#fff; cursor:pointer; position:relative; flex-shrink:0;
margin:0; padding:0; }'
    + '#dualSummaryCheck:checked { background:#d97706; }'
    + '#dualSummaryCheck:checked::after { content:""; position:absolute;
left:3px; top:0px; width:3px; height:7px; border:solid #fff;
border-width:0 2px 2px 0; transform:rotate(45deg); }'
    + '#dualSummaryCheck:hover { border-color:#996600; }'
    + '#dualSummaryLabel { font-size:7px; color:#996600;
letter-spacing:1px; text-transform:uppercase; }'
    + '#chatFooter { display:flex; justify-content:space-between;
align-items:center; gap:8px; }'
    + '#chatStatus { font-size:8px; color:#999; flex:1;
white-space:nowrap; overflow:hidden; text-overflow:ellipsis; }'
    + '#chatStatus.ok { color:#3a8f3a; }'
    + '#chatStatus.warn { color:#cc8800; }'
    + '#chatStatus.err { color:#cc3333; }'
    + '#chatBtns { display:flex; gap:6px; flex-shrink:0; }'
    + '#chatStats, #chatExport, #chatFullscreen { background:#fff;
border:2px solid #ccc; color:#666; padding:7px 12px; font-family:' + PF +
'; font-size:8px; cursor:pointer; border-radius:4px; transition:all 0.2s
ease; display:inline-flex; align-items:center; gap:4px;
white-space:nowrap; ' + CRISP + ' }'
    + '#chatStats:hover, #chatExport:hover, #chatFullscreen:hover {
background:#e8f4ff; color:#0077aa; border-color:#0077aa; }'
    + '#chatStats.active { background:#0077aa; color:#fff;
border-color:#0077aa; }'
    + '#chatFullscreen.active { background:#005580; color:#fff;
border-color:#005580; }'
    + '#buddyFrame.fullscreen { position:fixed !important; top:0
```

```
!important; left:0 !important; width:100% !important; height:100%
!important; z-index:99998 !important; border-radius:0 !important;
margin:0 !important; padding:16px !important; box-sizing:border-box
!important; overflow-y:auto !important; }'
    + '#buddyFrame.fullscreen #chatLog { max-height:calc(100vh - 380px)
!important; }'
    + '#topic-body.helikon-fullscreen { position:fixed !important; top:0
!important; left:0 !important; width:100% !important; height:100%
!important; z-index:99997 !important; overflow-y:auto !important;
background:#f8f8f8 !important; box-sizing:border-box !important;
padding:0 !important; margin:0 !important; }'
    + '#aiStats { display:none; margin-top:10px; padding:14px; border:2px
solid #ddd; background:#f0f0f0; border-radius:6px; font-size:10px;
line-height:2.0; color:#333; }'
    + '#aiStats.visible { display:block; }'
    + '.statsTitle { font-size:8px; color:#0077aa; letter-spacing:1px;
text-transform:uppercase; margin:10px 0 6px 0; padding:0 0 4px 0;
border-bottom:1px solid #ddd; }'
    + '.statsTitle:first-child { margin-top:0; }'
    + '.statRow { display:flex; justify-content:space-between;
padding:2px 0; }'
    + '.statLabel { color:#666; }'
    + '.statValue { color:#333; text-align:right; }'
    + '.energyBar { display:inline-block; height:6px; background:#4caf50;
margin-left:6px; vertical-align:middle; border-radius:3px; }'
    + '.scoreBox { text-align:center; padding:12px; margin-bottom:8px;
border:2px solid #0077aa; border-radius:6px; background:#e8f4ff; }'
    + '.scoreNum { font-size:16px; color:#0077aa; display:block;
margin:4px 0; }'
    + '.scoreLabel { font-size:8px; color:#666; text-transform:uppercase;
letter-spacing:1px; }'
    + '.scoreGrade { font-size:9px; margin-top:4px; }'
    + '.statIcon { display:inline-block; width:14px; text-align:center;
margin-right:3px; font-style:normal; }'
    + '.statSection { margin-top:6px; padding:8px 10px; border:1px solid
#ddd; border-radius:4px; background:#f8f8f8; }'
    + '.statSection .statsTitle { margin-top:0; }'
    + '.statProvName { font-size:9px; display:inline-block; padding:2px
6px; border-radius:3px; margin-left:4px; }'
    + '.statProvName.local { background:#e8ffe8; color:#3a8f3a;
border:1px solid #3a8f3a; }'
    + '.statProvName.claude { background:#fff8e8; color:#d97706;
border:1px solid #d97706; }'
    + '#mtOverlay { position:fixed; top:0; left:0; width:100%;
height:100%; background:rgba(0,0,0,0.7); z-index:99999;
justify-content:center; align-items:center; }'
    + '#mtOverlayInner { background:#fff; border:4px solid #d97706;
border-radius:16px; max-width:700px; width:90%; max-height:85vh;
overflow-y:auto; padding:40px; position:relative; box-shadow:0 0 60px
rgba(217,119,6,0.3); animation:overlayIn 0.4s ease; }'
```

```
    + '#mtOverlayInner::before { content:""; position:absolute;
top:-20px; left:50%; transform:translateX(-50%); width:0; height:0;
border-left:20px solid transparent; border-right:20px solid transparent;
border-bottom:20px solid #d97706; }'
    + '@keyframes overlayIn { 0% { transform:scale(0.8); opacity:0; }
100% { transform:scale(1); opacity:1; } }'
    + '#mtOverlayClose { position:absolute; top:14px; right:18px;
font-size:18px; color:#999; cursor:pointer; padding:4px 8px;
transition:color 0.2s ease; }'
    + '#mtOverlayClose:hover { color:#d97706; }'
    + '#mtOverlayContent { font-family:' + PF + '; font-size:11px;
line-height:2.2; color:#555; ' + CRISP + ' }'
    + '#mtOverlayContent .mtTitle { font-size:14px; color:#d97706;
margin:0 0 16px 0; padding:0 0 12px 0; border-bottom:2px solid #f0d0a0;
text-align:center; }'
    + '#mtOverlayContent .mtQuote { font-size:14px; color:#d97706;
text-align:center; margin:20px 0; padding:20px; border:2px dashed
#f0d0a0; border-radius:12px; background:#fffbf0; }'
    + '#mtOverlayContent .mtSection { margin-top:16px; padding-top:12px;
border-top:1px solid #eee; }'
    + '#mtOverlayContent .mtFact { padding:6px 0 6px 12px;
border-left:2px solid #d97706; margin:8px 0; }'
    + '.tts-play-btn { background:#fff; border:1px solid #ddd;
color:#999; padding:3px 8px; cursor:pointer; font-size:8px;
margin-left:8px; font-family:' + PF + '; display:none; border-radius:3px;
transition:all 0.2s ease; }'
    + '.tts-play-btn:hover { border-color:#0077aa; color:#0077aa; }'
    + '.tts-play-btn.playing { border-color:#0077aa; color:#0077aa;
display:inline-block !important; box-shadow:0 0 6px rgba(0,119,170,0.3);
}'
    + '.tts-active .tts-play-btn { display:inline-block; }'
    + '.chatMsg.call-status { background:transparent !important;
border:none !important; text-align:center !important; padding:6px 10px
!important; margin:4px auto !important; max-width:100% !important;
font-size:9px !important; opacity:0.8; }'
    + '.chatMsg.call-status::before { content:none !important; }'
    + '#mtTeam { margin-top:20px; padding-top:16px; border-top:2px solid
#f0d0a0; }'
    + '#mtTeam .mtTeamTitle { font-size:11px; color:#d97706;
text-align:center; margin-bottom:16px; letter-spacing:1px; }'
    + '.mtCredit { display:flex; align-items:center; gap:14px;
padding:12px; margin:10px 0; border:2px solid #eee; border-radius:8px;
background:#fff; transition:all 0.3s ease; }'
    + '.mtCredit:hover { border-color:#d97706; background:#fffbf0; }'
    + '.mtCreditIcon { font-size:28px; flex-shrink:0; width:50px;
text-align:center; }'
    + '.mtCreditName { font-size:10px; color:#d97706; display:block;
margin-bottom:2px; letter-spacing:0.5px; }'
    + '.mtCreditRole { font-size:8px; color:#888; display:block; }'
    + '.mtCreditHint { font-size:8px; color:#bb8833; margin-top:4px;
```

```
display:block; font-style:italic; }'
    + '.mtFeatureList { margin:12px 0; padding:0; }'
    + '.mtFeatureItem { display:flex; align-items:flex-start; gap:8px;
padding:4px 0; font-size:9px; color:#666; }'
    + '.mtFeatureItem::before { content:"\\25B8"; color:#d97706;
flex-shrink:0; }'
    + '@keyframes rainbowGlow { 0% { text-shadow:0 0 12px #ff0000; } 14%
{ text-shadow:0 0 12px #ff8800; } 28% { text-shadow:0 0 12px #ffff00; }
42% { text-shadow:0 0 12px #00ff00; } 57% { text-shadow:0 0 12px #0088ff;
} 71% { text-shadow:0 0 12px #4400ff; } 85% { text-shadow:0 0 12px
#ff00ff; } 100% { text-shadow:0 0 12px #ff0000; } }'
    + '.rainbowIcon { animation:rainbowGlow 3s linear infinite;
display:inline-block; }'
    + '@keyframes sparkle { 0%,100% { opacity:1; transform:scale(1); }
50% { opacity:0.6; transform:scale(1.2); } }'
    + '.sparkleIcon { animation:sparkle 2s ease-in-out infinite;
display:inline-block; }'
    + '@media (max-width:768px) {'
    + '  #buddyFrame { padding:10px; }'
    + '  #buddyChat { flex-direction:column; }'
    + '  #buddySide { width:100%; flex-direction:row; gap:10px;
position:static; padding:8px 10px; }'
    + '  #buddyChatImg { width:80px; height:80px; }'
    + '  #buddyBubble { max-width:60%; font-size:8px; margin-bottom:0; }'
    + '  #buddyNameTag { display:none; }'
    + '  #buddyDragHandle { display:none; }'
    + '  #chatSend { min-width:100px; padding:18px 24px; font-size:14px;
}'
    + '  #chatInput { font-size:10px; }'
    + '  .chatMsg { font-size:10px; max-width:95%; }'
    + '  #llmSwitch { gap:4px; }'
    + '  .llmBtn { padding:6px 8px; font-size:7px; flex:1;
text-align:center; }'
    + '  #chatBtns { gap:4px; }'
    + '  #chatStats, #chatExport, #chatFullscreen { padding:6px 8px;
font-size:7px; }'
    + '}';

    var styleEl = document.createElement('style');
    styleEl.type = 'text/css';
    styleEl.appendChild(document.createTextNode(css));
    document.head.appendChild(styleEl);

    /* === CONFIG === */
    var SHORT_LEN    = 100;
    var ENERGY_PER_TOKEN = 0.000000015;
    var DUAL_MAX_ROUNDS = 6;
    var DUAL_MIN_ROUNDS = 3;
    var DUAL_CEIL_ROUNDS = 20;
    var AT = String.fromCharCode(64);
```

```
    /* --- firnCore customization --- v2.6.8: Such-Konfiguration */
    var SEARCH_LIMIT        = 5;
    var SEARCH_FILE_LIMIT   = 3;
    var SEARCH_FETCH        = 25;
    var SEARCH_INCLUDE_USER = false;
    var SEARCH_MIN_WORD     = 3;
    var CTX_TOP_PAGES       = 2;
    var CTX_CHARS           = 1500;
    var CTX_LEAD            = 300;
    var SEARCH_DEBUG        = true;

    /* === PROVIDER STATE === */
    var providers = new Object();
    providers.p1 = new Object();
    providers.p1.url = ''; providers.p1.key = ''; providers.p1.model =
'default';
    providers.p1.name = 'Local'; providers.p1.type = 'openai';
providers.p1.ready = false;
    providers.p1.mentions = ['bwo', 'local', 'nima', 'concierge',
'llama'];
    providers.p1.sysMsg = ''; providers.p1.staticInfo = '';
providers.p1.configErrors = [];
    providers.p2 = new Object();
    providers.p2.url = ''; providers.p2.key = ''; providers.p2.model =
'claude-sonnet-4-20250514';
    providers.p2.name = 'Claude'; providers.p2.type = 'anthropic';
providers.p2.ready = false;
    providers.p2.mentions = ['claude', 'anthropic', 'sonnet'];
    providers.p2.sysMsg = ''; providers.p2.staticInfo = '';
providers.p2.configErrors = [];

    var activeProvider = 'p1';
    var dualMode = false;
    var dualRound = 0;
    var dualChaining = false;
    var dualAbortFlag = false;
    var dualSummaryEnabled = false;
    var currentXhr = null;

    /* === SESSION STATS === */
    var ss = new Object();
    ss.start = Date.now(); ss.reqs = 0; ss.errors = 0;
    ss.tokIn = 0; ss.tokOut = 0; ss.chIn = 0; ss.chOut = 0;
    ss.timeMs = 0; ss.lastMs = 0; ss.provider = ''; ss.model = '';
    var ssP1 = new Object(); ssP1.reqs = 0; ssP1.errors = 0; ssP1.tokIn =
0; ssP1.tokOut = 0; ssP1.timeMs = 0; ssP1.lastMs = 0;
    var ssP2 = new Object(); ssP2.reqs = 0; ssP2.errors = 0; ssP2.tokIn =
0; ssP2.tokOut = 0; ssP2.timeMs = 0; ssP2.lastMs = 0;
```

```
    /* === DOM === */
    var log        = document.getElementById('chatLog');
    var input      = document.getElementById('chatInput');
    var btn        = document.getElementById('chatSend');
    var status     = document.getElementById('chatStatus');
    var avatar     = document.getElementById('buddyChatImg');
    var bubble     = document.getElementById('buddyBubble');
    var exportBtn  = document.getElementById('chatExport');
    var statsBtn   = document.getElementById('chatStats');
    var fsBtn      = document.getElementById('chatFullscreen');
    var buddyFrame = document.getElementById('buddyFrame');
    var statsDiv   = document.getElementById('aiStats');
    var llmBtn1    = document.getElementById('llmBtn1');
    var llmBtn2    = document.getElementById('llmBtn2');
    var llmBtnDual = document.getElementById('llmBtnDual');
    var buddySide   = document.getElementById('buddySide');
    var dragHandle = document.getElementById('buddyDragHandle');
    var dualInfo   = document.getElementById('dualInfo');
    var dualInfoText = document.getElementById('dualInfoText');
    var dualRndVal = document.getElementById('dualRndVal');
    var dualRndDown = document.getElementById('dualRndDown');
    var dualRndUp  = document.getElementById('dualRndUp');
    var dualSummaryCheck = document.getElementById('dualSummaryCheck');
    var mtOverlay  = document.getElementById('mtOverlay');
    var mtClose    = document.getElementById('mtOverlayClose');
    var mtContent  = document.getElementById('mtOverlayContent');
    if (!log || !input || !btn) return;

    input.placeholder = 'Schreib mir...';
    log.innerHTML = '';

    /* v2.2: Right-click chatLog = copy to clipboard */
    log.addEventListener('contextmenu', function(e) {
        var msgs = log.getElementsByClassName('chatMsg');
        if (msgs.length === 0) return;
        e.preventDefault();
        var text = '';
        var i;
        for (i = 0; i < msgs.length; i++) {
            var el = msgs[i];
            if (el.className.indexOf('typing') > -1) continue;
            var role = el.className.indexOf('user') > -1 ? 'USER' :
'BUDDY';
            if (el.className.indexOf('system') > -1) role = 'SYS';
            if (el.className.indexOf('fact') > -1) role = 'FACT';
            if (el.className.indexOf('dual-chain') > -1) role = 'DUAL';
            text += '[' + role + '] ' + (el.textContent || el.innerText
|| '') + '\n';
        }
        if (navigator.clipboard && navigator.clipboard.writeText) {
```

```
            navigator.clipboard.writeText(text).then(function() {
showBubble('Chat kopiert!', 2000); });
        } else {
            var ta = document.createElement('textarea');
            ta.value = text; ta.style.position = 'fixed'; ta.style.left =
'-9999px';
            document.body.appendChild(ta); ta.select();
            try { document.execCommand('copy'); showBubble('Chat
kopiert!', 2000); } catch(ex) {}
            document.body.removeChild(ta);
        }
    });

    var GIFS = new Object();
    GIFS.idle    = '/@api/deki/files/410/=buddy_idle.gif';
    GIFS.sleep   = '/@api/deki/files/412/=buddy_sleeping.gif';
    GIFS.happy   = '/@api/deki/files/409/=buddy_happy.gif';
    GIFS.error   = '/@api/deki/files/408/=buddy_error.gif';
    GIFS.loading = '/@api/deki/files/411/=buddy_loading.gif';

    var history = [];
    var busy = false;
    /* v2.6.4: sysMsg/staticInfo sind jetzt PRO PROVIDER
(providers.p1.sysMsg etc.) */
    var teamImgUrl = '';
    var statsVisible = false;
    var bubbleTimer = null;

    var pageId = 0;
    var userName = 'Anonymous User';
    var pagePath = '/';
    if (typeof Deki !== 'undefined') {
        if (Deki.PageId) pageId = Deki.PageId;
        if (Deki.UserName) userName = Deki.UserName;
        if (Deki.PageTitle) pagePath = Deki.PageTitle;
    }
    if (typeof YAHOO !== 'undefined' && YAHOO.Deki &&
YAHOO.Deki.PagePath) {
        pagePath = YAHOO.Deki.PagePath;
    }

    /* === DRAGGABLE BUDDY === */
    var isDragging = false;
    var dragOffX = 0, dragOffY = 0;

    if (dragHandle && buddySide) {
        dragHandle.addEventListener('mousedown', function(e) {
            e.preventDefault();
            isDragging = true;
            var rect = buddySide.getBoundingClientRect();
```

```
            dragOffX = e.clientX - rect.left;
            dragOffY = e.clientY - rect.top;
            buddySide.style.left = rect.left + 'px';
            buddySide.style.top = rect.top + 'px';
            buddySide.style.width = rect.width + 'px';
            buddySide.className = 'dragging';
        });
        dragHandle.addEventListener('touchstart', function(e) {
            var touch = e.touches[0];
            isDragging = true;
            var rect = buddySide.getBoundingClientRect();
            dragOffX = touch.clientX - rect.left;
            dragOffY = touch.clientY - rect.top;
            buddySide.style.left = rect.left + 'px';
            buddySide.style.top = rect.top + 'px';
            buddySide.style.width = rect.width + 'px';
            buddySide.className = 'dragging';
        });
        document.addEventListener('mousemove', function(e) {
            if (!isDragging) return;
            buddySide.style.left = (e.clientX - dragOffX) + 'px';
            buddySide.style.top = (e.clientY - dragOffY) + 'px';
        });
        document.addEventListener('touchmove', function(e) {
            if (!isDragging) return;
            var touch = e.touches[0];
            buddySide.style.left = (touch.clientX - dragOffX) + 'px';
            buddySide.style.top = (touch.clientY - dragOffY) + 'px';
        });
        document.addEventListener('mouseup', function() { isDragging =
false; });
        document.addEventListener('touchend', function() { isDragging =
false; });
        dragHandle.addEventListener('dblclick', function() {
            buddySide.className = '';
            buddySide.style.left = '';
            buddySide.style.top = '';
            buddySide.style.width = '';
            showBubble('Bin wieder da!', 2000);
        });
    }

    /* === LOGGING === */
    function logToServer(type, extra) {
        try {
            var payload = new Object();
            payload.type = type; payload.user = userName; payload.page =
pagePath;
            payload.provider = (extra && extra.provider) ? extra.provider
: '-';
```

```
            payload.model = (extra && extra.model) ? extra.model : '-';
            payload.tokens = (extra && extra.tokens) ? extra.tokens : 0;
            payload.latency = (extra && extra.latency) ? extra.latency :
0;
            payload.message = (extra && extra.message) ? extra.message :
'';
            var xhr = new XMLHttpRequest();
            xhr.open('POST', '/helikon-logger.php', true);
            xhr.setRequestHeader('Content-Type', 'application/json');
            xhr.send(JSON.stringify(payload));
        } catch (e) {}
    }

    /* === HTML TAG HELPERS === */
    var T = String.fromCharCode(60);
    var TC = String.fromCharCode(60) + '/';
    var G = String.fromCharCode(62);

    function tag(el, cls, inner) { return T + el + ' class="' + cls + '"'
+ G + inner + TC + el + G; }
    function tagS(el, cls, style, inner) { return T + el + ' class="' +
cls + '" style="' + style + '"' + G + inner + TC + el + G; }
    function makeMsg(role, content) { var m = new Object(); m.role =
role; m.content = content; return m; }
    function estimateTokens(text) { return Math.ceil(text.length / 4); }

    /* === UTILITIES === */
    function fetchText(url, callback) {
        var xhr = new XMLHttpRequest();
        xhr.open('GET', url, true);
        xhr.onload = function() { callback(xhr.status === 200 ?
xhr.responseText.replace(/^\s+|\s+$/g, '') : ''); };
        xhr.onerror = function() { callback(''); };
        xhr.send();
    }

   function fetchAllProperties(callback) {
       if (!pageId) { callback(new Object()); return; }
       var xhr = new XMLHttpRequest();
       xhr.open('GET', '/helikon-proxy.php?config=' + pageId, true);
       xhr.onload = function() {
           var props = new Object();
           if (xhr.status === 200) { try { props =
JSON.parse(xhr.responseText); } catch (e) {} }
           callback(props);
       };
       xhr.onerror = function() { callback(new Object()); };
       xhr.send();
   }
```

```
    /* === MINDTOUCH API LAYER === */
    var mtConfig = new Object();
    mtConfig.base = '/@api/deki'; mtConfig.key = ''; mtConfig.ready =
false;

    function mtApi(endpoint, callback) {
        var url = mtConfig.base + endpoint;
        var xhr = new XMLHttpRequest();
        xhr.open('GET', url, true);
        if (mtConfig.key) xhr.setRequestHeader('Authorization', 'Bearer '
+ mtConfig.key);
        xhr.onload = function() {
            if (xhr.status === 200 || xhr.status === 304) {
                try {
                    var parser = new DOMParser();
                    var xml = parser.parseFromString(xhr.responseText,
'text/xml');
                    callback(null, xml);
                } catch (e) { callback('Parse-Fehler', null); }
            } else { callback('HTTP ' + xhr.status, null); }
        };
        xhr.onerror = function() { callback('Nicht erreichbar', null); };
        xhr.send();
    }

    function mtSubpages(mPath, callback) {
        var encoded = encodeURIComponent(mPath.replace(/\//g, '%2F'));
        mtApi('/pages/=' + encoded + '/subpages', function(err, xml) {
            if (err) { callback(err, []); return; }
            var results = [];
            try {
                var pages = xml.getElementsByTagName('page.subpage');
                if (pages.length === 0) pages =
xml.getElementsByTagName('page');
                var i;
                for (i = 0; i < pages.length; i++) {
                    var entry = new Object();
                    entry.id = pages[i].getAttribute('id') || '';
                    var titleEl = pages[i].getElementsByTagName('title');
                    var pathEl = pages[i].getElementsByTagName('path');
                    entry.title = titleEl.length > 0 ?
titleEl[0].textContent : '';
                    entry.path = pathEl.length > 0 ?
pathEl[0].textContent : '';
                    if (entry.title) results.push(entry);
                }
            } catch (e) {}
            callback(null, results);
        });
    }
```

```
    function mtPageContent(pageIdArg, callback) { mtApi('/pages/' +
pageIdArg + '/contents', callback); }

    function mtPageProps(pageIdArg, callback) {
        mtApi('/pages/' + pageIdArg + '/properties', function(err, xml) {
            if (err) { callback(err, new Object()); return; }
            var props = new Object();
            try {
                var nodes = xml.getElementsByTagName('property');
                var i;
                for (i = 0; i < nodes.length; i++) {
                    var name = nodes[i].getAttribute('name') || '';
                    var contents =
nodes[i].getElementsByTagName('contents');
                    var val = '';
                    if (contents.length > 0 && contents[0].textContent)
val = contents[0].textContent.replace(/^\s+|\s+$/g, '');
                    props[name.replace('urn:custom.mindtouch.com#', '')]
= val;
                }
            } catch (e) {}
            callback(null, props);
        });
    }

    window.helikonApi = new Object();
    window.helikonApi.mt = mtApi;
    window.helikonApi.mtSubpages = mtSubpages;
    window.helikonApi.mtPageContent = mtPageContent;
    window.helikonApi.mtPageProps = mtPageProps;
    window.helikonApi.mtConfig = mtConfig;
    window.helikonApi.providers = providers;

    /* ============================================================
     * --- firnCore customization ---
     * v2.6.8: WIKI SEARCH — /site/query (bestguess)
     * Gleiche Engine wie die MindTouch-Hauptsuche. /site/search
     * crasht serverseitig bei Datei-Treffern (HTTP 500).
     * ============================================================ */
    function sdbg(m) {
        if (SEARCH_DEBUG && window.console) console.log('[Buddy Search
v2.6.8] ' + m);
    }

    var SEARCH_STOP = new Array(
        /* DE: Fragen / Aufforderungen */
        'kennst','kennt','weisst','wei\
u00DFt','was','ist','sind','war','waren','wer','wie','wo','wann','warum','wieso'
        'kannst','koennte','k\
```

```
u00F6nnte','kann','soll','sollte','will','willst','moechte','m\
u00F6chte','hast','habt','haben','hat','gibt','gib','gibts',

'sagen','sag','zeig','zeige','zeigen','finde','finden','suche','suchen','such','
u00E4hl','erzaehle','erz\u00E4hle','erklaer','erkl\
u00E4r','erklaere','erkl\u00E4re','nenn','nenne',
        /* DE: Artikel / Pronomen / Praepositionen / Fuellwoerter */

'ein','eine','einen','einem','einer','eines','der','die','das','den','dem','des'

'und','oder','aber','ich','mich','mir','du','dir','es','er','sie','wir','uns','i

'von','vom','zu','zum','zur','im','in','am','an','auf','aus','bei','mit','nach',
u00FCr','ohne','um','bis','seit',

'bitte','mal','denn','auch','noch','schon','ja','nein','alle','allen','alles','e
u00FCber','hier','dort','gerne','gern',

'info','infos','information','informationen','irgendwas','irgendwelche',
        'danke','merci','ok','okay','thx','super','prima','genau',
        /* EN */

'do','does','you','know','what','is','are','was','were','the','a','an','can','co

'find','search','for','tell','about','where','when','who','how','which','of','on

'give','any','some','all','please','have','has','there','anything','thanks'
    );

    /* Alles ausser Buchstaben/Ziffern (inkl. Umlaute) wird Leerzeichen.
     * Als RegExp aus String gebaut: Regex-Literale mit Anfuehrungszeichen
     * brechen den MindTouch-JEM-Parser (ganzes Script faellt aus). */
    var NON_WORD_RX = new RegExp('[^0-9a-zA-Z\u00C0-\u024F]+', 'g');

    function isSearchStop(w) {
        var i;
        for (i = 0; i < SEARCH_STOP.length; i++) { if (w ===
SEARCH_STOP[i]) return true; }
        return false;
    }

    /* Prompt -> Suchbegriffe. Liefert leeren String wenn nur
Fuellwoerter uebrig sind. */
    function cleanSearchQuery(msg) {
        var q = ('' + msg)
            .replace(NON_WORD_RX, ' ')
            .replace(/\s+/g, ' ')
            .replace(/^\s+|\s+$/g, '');
        if (!q) return '';
        var words = q.split(' ');
```

```
        var kept = new Array();
        var i;
        for (i = 0; i < words.length; i++) {
            var w = words[i].toLowerCase();
            if (w.length > 1 && !isSearchStop(w)) kept.push(words[i]);
        }
        return kept.join(' ');
    }

    function searchWords(q) {
        var parts = q.toLowerCase().split(/\s+/);
        var seen = new Object();
        var out = new Array();
        var i;
        for (i = 0; i < parts.length; i++) {
            var w = parts[i];
            if (w.length >= SEARCH_MIN_WORD && !seen[w]) { seen[w] =
true; out.push(w); }
        }
        return out;
    }

    function isSearchJunkPath(p) {
        p = '' + (p || '');
        if (p.indexOf('Template:') === 0) return true;
        if (p.indexOf('Special:') === 0) return true;
        if (!SEARCH_INCLUDE_USER && p.indexOf('User:') === 0) return true;
        if (p.indexOf('[object') !== -1) return true;
        return false;
    }

    function isSearchJunkFile(fn) {
        fn = ('' + (fn || '')).toLowerCase();
        if (fn.indexOf('mindtouch.import') !== -1) return true;
        if (fn.indexOf('mindtouch#') !== -1) return true;
        if (fn === 'element.js') return true;
        if (fn.length < 20 && fn.indexOf('.js') === fn.length - 3) return
true;
        if (fn.length < 20 && fn.indexOf('.css') === fn.length - 4)
return true;
        return false;
    }

    /* Ausschnitt rund um den ersten Treffer (statt Anfang der Seite) */
    function textWindow(text, words, maxLen, lead) {
        var plain = ('' + (text || '')).replace(/[\s\u00A0]+/g, '
').replace(/^\s+|\s+$/g, '');
        if (!plain) return '';
        if (plain.length <= maxLen) return plain;
        var lower = plain.toLowerCase();
```

```
        var idx = -1;
        var i;
        for (i = 0; i < words.length; i++) {
            var p = lower.indexOf(words[i]);
            if (p !== -1 && (idx === -1 || p < idx)) idx = p;
        }
        var start = 0;
        if (idx > lead) {
            start = idx - lead;
            var sp = plain.indexOf(' ', start);
            if (sp !== -1 && sp < idx) start = sp + 1;
        }
        var end = start + maxLen;
        if (end > plain.length) {
            end = plain.length;
            start = Math.max(0, end - maxLen);
        }
        return (start > 0 ? '\u2026 ' : '') + plain.substring(start, end)
+ (end < plain.length ? ' \u2026' : '');
    }

    function searchWiki(query, words, callback) {
        var url = '/@api/deki/site/query?q=' + encodeURIComponent(query)
                + '&parser=bestguess&limit=' + SEARCH_FETCH +
'&dream.out.format=json';
        sdbg('GET ' + url);
        var xhr = new XMLHttpRequest();
        xhr.open('GET', url, true);
        xhr.onload = function() {
            var pages = new Array();
            var files = new Array();
            if (xhr.status === 200) {
                try {
                    var data = JSON.parse(xhr.responseText);
                    sdbg('count=' + (data[AT + 'count'] || 0) + '
parsedQuery=' + (data.parsedQuery || ''));
                    var arr = data.result;
                    if (!arr) arr = new Array();
                    else if (typeof arr.length !== 'number') { var one =
arr; arr = new Array(); arr.push(one); }
                    var seenP = new Object();
                    var seenF = new Object();
                    var i;
                    for (i = 0; i < arr.length; i++) {
                        var r = arr[i] || new Object();
                        var pi = r.page || new Object();
                        var ppath = '' + (pi.path || '');
                        if (isSearchJunkPath(ppath)) continue;

                        if (r.type === 'file') {
```

```
                            if (files.length >= SEARCH_FILE_LIMIT)
continue;
                            var fname = r.title || '';
                            var fkey = r.id || fname;
                            if (!fname || isSearchJunkFile(fname) ||
seenF[fkey]) continue;
                            seenF[fkey] = true;
                            var fe = new Object();
                            fe.filename  = fname;
                            fe.mime      = r.mime || '';
                            fe.pageTitle = pi.title || '';
                            fe.pagePath  = ppath;
                            files.push(fe);
                        } else if (r.type === 'page' || r.type ===
'comment') {
                            if (pages.length >= SEARCH_LIMIT ||
seenP[ppath]) continue;
                            seenP[ppath] = true;
                            var mId = ('' + (r['uri.track'] ||
'')).match(/pageid=(\d+)/);
                            var pe = new Object();
                            pe.id      = mId ? mId[1] : (r.type ===
'page' ? (r.id || '') : '');
                            pe.title   = (r.type === 'page' ? r.title :
pi.title) || ppath;
                            pe.path    = ppath;
                            pe.uri     = r.uri || '';
                            pe.summary = textWindow(r.content || '',
words, 300, 80);
                            pe.source  = r.type;
                            pages.push(pe);
                        }
                    }
                } catch (e) { sdbg('Parse-Fehler: ' + e); }
            } else { sdbg('HTTP ' + xhr.status); }
            sdbg('Kontext: pages=' + pages.length + ' files=' +
files.length);
            callback(pages, files);
        };
        xhr.onerror = function() { sdbg('nicht erreichbar'); callback(new
Array(), new Array()); };
        xhr.send();
    }

    /* ============================================================
     * v2.6.8: fetchPageText — Volltext, Ausschnitt rund um Treffer
     * ============================================================ */
    function fetchPageText(pgId, words, callback) {
        var url = '/@api/deki/pages/' + pgId +
'/contents?dream.out.format=json';
```

```
        var xhr = new XMLHttpRequest();
        xhr.open('GET', url, true);
        xhr.onload = function() {
            if (xhr.status === 200) {
                try {
                    var data = JSON.parse(xhr.responseText);
                    var html = data.body || data['#text'] || '';
                    if (!html && data.content) html = data.content;
                    /* body kann Array sein (Seite + TOC) oder Objekt mit
#text */
                    if (typeof html !== 'string') {
                        if (typeof html.length === 'number' &&
html.length > 0) html = html[0];
                        if (html && typeof html === 'object') html =
html['#text'] || '';
                        if (typeof html !== 'string') html = '';
                    }
                    var tmp = document.createElement('div');
                    tmp.innerHTML = html;
                    var text = (tmp.textContent || tmp.innerText || '');
                    callback(textWindow(text, words, CTX_CHARS,
CTX_LEAD));
                } catch(e) { callback(''); }
            } else { callback(''); }
        };
        xhr.onerror = function() { callback(''); };
        xhr.send();
    }

    /* ============================================================
     * v2.6.8: buildSearchContext — Seiten + Dateien
     * ============================================================ */
    function buildSearchContext(results, files, words, doneCb) {
        if ((!results || results.length === 0) && (!files || files.length
=== 0)) { doneCb(''); return; }
        var topN = Math.min(results.length, CTX_TOP_PAGES);
        var pending = topN;
        var fullTexts = new Array();
        var i;
        for (i = 0; i < topN; i++) fullTexts.push('');

        function finish() {
            var ctx = '\n\n[Relevante Wiki-Seiten gefunden — sinngemass
zusammenfassen, nicht woertlich zitieren]\n';
            var ri;
            for (ri = 0; ri < results.length; ri++) {
                var r = results[ri];
                ctx += '\n' + (ri + 1) + '. ' + r.title + ' (/' + r.path
+ ')';
                if (ri < topN && fullTexts[ri]) {
```

```
                    ctx += '\n   Inhalt: ' + fullTexts[ri];
                } else if (r.summary) {
                    ctx += '\n   Auszug: ' + r.summary;
                }
                ctx += '\n';
            }
            if (files && files.length > 0) {
                ctx += '\nPassende Dateien:\n';
                var fi;
                for (fi = 0; fi < files.length; fi++) {
                    var f = files[fi];
                    ctx += '- ' + f.filename;
                    if (f.pageTitle) ctx += ' (auf Seite ' + f.pageTitle
+ ', /' + f.pagePath + ')';
                    ctx += '\n';
                }
            }
            ctx += '\nWenn du auf eine dieser Seiten verweist, nenne den
Pfad als klickbaren Link (z.B. /Bereiche/MUSEUM/Computer/...).\n';
            doneCb(ctx);
        }

        if (topN === 0) { finish(); return; }

        for (i = 0; i < topN; i++) {
            (function(idx, entry) {
                if (!entry.id) {
                    pending--;
                    if (pending <= 0) finish();
                    return;
                }
                fetchPageText(entry.id, words, function(text) {
                    fullTexts[idx] = text;
                    pending--;
                    if (pending <= 0) finish();
                });
            })(i, results[i]);
        }
    }

    function isGreeting(msg) {
        var lower = msg.toLowerCase().replace(new RegExp(AT + '\\w+\\s*',
'g'), '').replace(/^\s+|\s+$/g, '');
        var greetings = new Array('hi','hallo','hey','moin','servus','gr\
u00FCezi','gr\u00FCzi','sal\u00FC','yo','guten tag','guten morgen','guten
abend','hoi');
        var i;
        for (i = 0; i < greetings.length; i++) {
            if (lower === greetings[i] || lower === greetings[i] + '!' ||
lower === greetings[i] + '.') return true;
```

```
        }
        return false;
    }

    /* === MENTION DETECTION === */
    function detectMention(text) {
        var mentionRx = new RegExp(AT + '(\\w+)');
        var match = text.toLowerCase().match(mentionRx);
        if (!match) return null;
        var name = match[1];
        var i;
        for (i = 0; i < providers.p1.mentions.length; i++) { if (name ===
providers.p1.mentions[i]) return 'p1'; }
        if (providers.p1.name && name ===
providers.p1.name.toLowerCase().split(' ')[0]) return 'p1';
        for (i = 0; i < providers.p2.mentions.length; i++) { if (name ===
providers.p2.mentions[i]) return 'p2'; }
        if (providers.p2.name && name ===
providers.p2.name.toLowerCase().split(' ')[0]) return 'p2';
        return null;
    }

    function stripMention(text) {
        return text.replace(new RegExp(AT + '\\w+\\s*'),
'').replace(/^\s+|\s+$/g, '');
    }

    /* === UI === */
    function setGif(name) { if (avatar) avatar.src = GIFS[name] ||
GIFS.idle; }

    function showBubble(txt, duration) {
        if (!bubble) return;
        if (bubbleTimer) clearTimeout(bubbleTimer);
        bubble.textContent = txt;
        bubble.className = 'visible';
        bubbleTimer = setTimeout(function() { bubble.className = ''; },
duration || 5000);
    }

    /* === LINK DETECTION === */
    var HASH = String.fromCharCode(35);

    function makeLink(p) { return T + 'a href="' + p + '"
target="_blank"' + G + p + TC + 'a' + G; }

    function extractFullPath(s, start) {
        if (start + 1 >= s.length) return null;
        var c1 = s.charAt(start + 1);
        if (!((c1 >= 'A' && c1 <= 'Z') || (c1 >= 'a' && c1 <= 'z') || (c1
```

```
>= '0' && c1 <= '9') || c1 === '_')) return null;
        var i = start + 1;
        while (i < s.length) {
            var c = s.charAt(i);
            if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') || (c >=
'0' && c <= '9') ||
                c === '_' || c === '-' || c === '/' || c === '.' || c ===
':' ||
                c === '?' || c === '=' || c === '&' || c === '%' || c ===
HASH) {
                i++;
            } else { break; }
        }
        if (i === start + 1) return null;
        var path = s.substring(start, i);
        while (path.length > 1 && '.!?,;'.indexOf(path.charAt(path.length
- 1)) !== -1) {
            path = path.substring(0, path.length - 1);
        }
        if (path.length < 2) return null;
        return path;
    }

    function linkifyReply(text) {
        var urlBlocks = [];
        var URLPH = '%%URL';
        var result = '';
        var i = 0;
        while (i < text.length) {
            if (text.substring(i, i + 8) === 'https://' ||
text.substring(i, i + 7) === 'http://') {
                var ue = i;
                while (ue < text.length && text.charAt(ue) !== ' ' &&
text.charAt(ue) !== '\n' && text.charAt(ue) !== '\t' && text.charAt(ue)
!== String.fromCharCode(60)) ue++;
                var url = text.substring(i, ue);
                while (url.length > 0 &&
'.!?,;)'.indexOf(url.charAt(url.length - 1)) !== -1) url =
url.substring(0, url.length - 1);
                var idx = urlBlocks.length;
                urlBlocks.push(T + 'a href="' + url + '" target="_blank"'
+ G + url + TC + 'a' + G);
                result += URLPH + idx + '%%';
                i += url.length;
            } else {
                result += text.charAt(i);
                i++;
            }
        }
        text = result;
```

```
        var out = '';
        var j = 0;
        while (j < text.length) {
            var ch = text.charAt(j);
            if (ch === '/') {
                var prev = j > 0 ? text.charAt(j - 1) : ' ';
                var prevOk = (prev === ' ' || prev === '\n' || prev ===
'\t' || prev === '(' ||
                              prev === '"' || prev === "'" || prev ===
',' || prev === ':' ||
                              prev === ';' || prev === '\r' || j === 0);
                if (prevOk) {
                    var fullPath = extractFullPath(text, j);
                    if (fullPath && fullPath.length > 1) {
                        var tail = '';
                        var fp = fullPath;
                        if (prev === '(' && fp.charAt(fp.length - 1) ===
')') {
                            fp = fp.substring(0, fp.length - 1);
                            tail = ')';
                        }
                        out += makeLink(fp) + tail;
                        j += fp.length + tail.length;
                        continue;
                    }
                }
            }
            out += ch;
            j++;
        }
        text = out;

        var ui;
        for (ui = 0; ui < urlBlocks.length; ui++) {
            text = text.replace(URLPH + ui + '%%', urlBlocks[ui]);
        }
        return text;
    }

    /* === MARKDOWN FORMATTER === */
    function formatReply(text) {
        var codeBlocks = [];
        text = text.replace(/```[\w]*\n?([\s\S]*?)```/g, function(m,
code) {
            var idx = codeBlocks.length;
            codeBlocks.push(T + 'pre' + G + T + 'code' + G +
code.replace(/&/g,'&amp;').replace(/</g,'&lt;').replace(/>/g,'&gt;') + TC
+ 'code' + G + TC + 'pre' + G);
            return '%%CODE' + idx + '%%';
```

```
        });
        text = text.replace(/`([^`]+)`/g, T + 'code' + G + '$1' + TC +
'code' + G);
        text = text.replace(/(?:^|\n)\*\*([^*]+)\*\*(?:\n|$)/g, '\n' + T
+ 'span class="mdH"' + G + '$1' + TC + 'span' + G + '\n');
        text = text.replace(/\*\*([^*]+)\*\*/g, T + 'strong' + G + '$1' +
TC + 'strong' + G);
        text = text.replace(/(?:^|\n)\* ([^\n]+)/g, '\n' + T + 'li' + G +
'$1' + TC + 'li' + G);
        text = text.replace(/((?:\s*<li>[\s\S]*?<\/li>\s*)+)/g, T + 'ul'
+ G + '$1' + TC + 'ul' + G);
        text = text.replace(/\n/g, T + 'br /' + G);
        var ci;
        for (ci = 0; ci < codeBlocks.length; ci++) { text =
text.replace('%%CODE' + ci + '%%', codeBlocks[ci]); }
        return text;
    }

    function addMsg(text, cls) {
        var div = document.createElement('div');
        div.className = 'chatMsg ' + cls;
        if (cls.indexOf('buddy') !== -1) {
            var linkBlocks = [];
            var linked = linkifyReply(text);
            var linkRx = new RegExp(T + 'a [^' + G + ']+' + G +
'[\\s\\S]*?' + TC + 'a' + G, 'g');
            linked = linked.replace(linkRx, function(m) { var idx =
linkBlocks.length; linkBlocks.push(m); return '%%LINK' + idx + '%%'; });
            var formatted = formatReply(linked);
            var li;
            for (li = 0; li < linkBlocks.length; li++) { formatted =
formatted.replace('%%LINK' + li + '%%', linkBlocks[li]); }
            div.innerHTML = formatted;
        } else { div.textContent = text; }
        log.appendChild(div);
        log.scrollTop = log.scrollHeight;
        return div;
    }

    function addProviderMsg(text, providerName, providerType) {
        var cls = 'buddy';
        if (providerType === 'anthropic') cls += ' claude';
        else cls += ' local';
        var div = document.createElement('div');
        div.className = 'chatMsg ' + cls;
        div.setAttribute('data-provider', providerType === 'anthropic' ?
'claude' : 'local');
        var provCls = providerType === 'anthropic' ? 'prov-claude' :
'prov-local';
        var badge = T + 'span class="msgProvider ' + provCls + '"' + G +
```

```
providerName + TC + 'span' + G + T + 'br /' + G;
        var linkBlocks = [];
        var linked = linkifyReply(text);
        var linkRx = new RegExp(T + 'a [^' + G + ']+' + G + '[\\s\\S]*?'
+ TC + 'a' + G, 'g');
        linked = linked.replace(linkRx, function(m) { var idx =
linkBlocks.length; linkBlocks.push(m); return '%%LINK' + idx + '%%'; });
        var formatted = formatReply(linked);
        var li;
        for (li = 0; li < linkBlocks.length; li++) { formatted =
formatted.replace('%%LINK' + li + '%%', linkBlocks[li]); }
        div.innerHTML = badge + formatted;
        log.appendChild(div);
        log.scrollTop = log.scrollHeight;
        return div;
    }

    function setStatus(txt, state) {
        if (!status) return;
        status.textContent = txt || '';
        status.className = state || '';
    }

    /* === LED MANAGEMENT === */
    function setLed(btnEl, state) {
        if (!btnEl) return;
        var led = btnEl.querySelector('.llmLed');
        if (led) led.className = 'llmLed ' + (state || '');
    }
    function updateLeds() {
        if (providers.p1.ready) { setLed(llmBtn1, ssP1.errors > 0 &&
ssP1.errors === ssP1.reqs ? 'err' : 'ok'); }
        else { setLed(llmBtn1, ''); }
        if (providers.p2.ready) { setLed(llmBtn2, ssP2.errors > 0 &&
ssP2.errors === ssP2.reqs ? 'err' : 'ok'); }
        else { setLed(llmBtn2, ''); }
    }

    /* === v2.6: PDF EXPORT === */
    function exportPdf() {
        var msgs = log.getElementsByClassName('chatMsg');
        if (msgs.length === 0) { showBubble('Nix zum Exportieren!',
3000); return; }
        var now = new Date();
        var dateStr = now.getFullYear() + '-' + ('0' + (now.getMonth() +
1)).slice(-2) + '-' + ('0' + now.getDate()).slice(-2);
        var timeStr = ('0' + now.getHours()).slice(-2) + ':' + ('0' +
now.getMinutes()).slice(-2);
        var html = T + '!DOCTYPE html' + G + T + 'html' + G + T + 'head'
+ G;
```

```
        html += T + 'meta charset="utf-8" /' + G;
        html += T + 'title' + G + 'Helikon Chat Export ' + dateStr + TC +
'title' + G;
        html += T + 'style' + G;
        html += '@import url("https://fonts.googleapis.com/
css2?family=Press+Start+2P");';
        html += 'body{font-family:"Courier
New",monospace;font-size:13px;color:#333;max-width:860px;margin:0
auto;padding:40px 24px;background:#f4f6f8;}';
        html +=
'.export-header{background:linear-gradient(135deg,#005580,#0077aa);color:#fff;pa
28px;border-radius:10px;margin-bottom:28px;}';
        html += '.export-header h1{margin:0 0 8px
0;font-size:15px;font-family:"Press Start 2P","Courier
New",monospace;letter-spacing:1px;}';
        html += '.export-header .meta{font-size:10px;opacity:0.8;}';
        html += '.msg{margin-bottom:14px;padding:14px
18px;border-radius:8px;line-height:1.7;word-wrap:break-word;}';
        html += '.msg-user{background:#e8e8e8;text-align:right;border:1px
solid #ddd;border-right:4px solid #888;}';
        html += '.msg-buddy{background:#fff;border:1px solid
#ddd;border-left:4px solid #0077aa;}';
        html += '.msg-buddy.claude{border-left-color:#d97706;}';
        html +=
'.msg-system{color:#999;font-size:11px;text-align:center;background:transparent;
        html +=
'.msg-fact{color:#0077aa;font-size:11px;background:#f0f8ff;border-left:3px
solid #0077aa;padding:8px 14px;border-radius:4px;}';
        html +=
'.msg-dual{color:#d97706;font-size:10px;text-align:center;background:#fffbf0;bor
dashed #d97706;padding:4px 10px;border-radius:4px;}';
        html += 'pre{background:#1a1a2e;color:#e8e8f0;padding:14px
16px;border-radius:6px;overflow-x:auto;font-size:12px;line-height:1.8;margin:10p
0;white-space:pre-wrap;word-wrap:break-word;border:1px solid #333355;}';
        html += 'code{background:#2a2a3e;color:#e8a87c;padding:2px
6px;border-radius:3px;font-size:12px;}';
        html += 'pre code{background:none;color:inherit;padding:0;}';
        html += 'a{color:#0077aa;text-decoration:none;border-bottom:1px
dotted #0077aa;}';
        html += 'strong{color:#005580;}';
        html += '.mtSearch{margin:10px 0 14px 0;border:2px solid
#0077aa;border-radius:8px;overflow:hidden;background:#f0f8ff;font-family:"Courie
New",monospace;}';
        html += '.mtSearch
.hdr{background:linear-gradient(135deg,#005580,#0077aa);color:#fff;padding:8px
14px;font-size:11px;display:flex;justify-content:space-between;}';
        html += '.mtSearch
.badge{background:rgba(255,255,255,0.2);padding:2px
8px;border-radius:10px;font-size:10px;}';
        html += '.mtSearch .sec{padding:5px
```

```
14px;background:#e0ecf4;font-size:10px;font-weight:bold;color:#005580;border-top
solid #cce0ee;}';
        html += '.mtSearch .item{padding:8px 14px;border-top:1px solid
#e0ecf4;font-size:10px;line-height:1.6;}';
        html += '.mtSearch .item
a{color:#0077aa;font-weight:bold;font-size:12px;display:block;}';
        html += '.mtSearch .item
.path{color:#999;font-size:9px;display:block;margin-top:2px;}';
        html += '.mtSearch .item
.snippet{color:#444;font-size:9px;display:block;margin-top:4px;border-left:2px
solid #b0d4e8;padding-left:6px;}';
        html += '.mtSearch .item
.meta{color:#777;font-size:9px;margin-top:3px;}';
        html += '.mtSearch .file-item{padding:6px 14px;border-top:1px
solid #e0ecf4;font-size:10px;display:flex;align-items:center;gap:10px;}';
        html += '.mtSearch
.file-thumb{width:48px;height:48px;object-fit:cover;border:1px solid
#cce0ee;border-radius:4px;}';
        html += '.mtSearch
.tag{display:inline-block;background:#0077aa;color:#fff;padding:2px
8px;border-radius:10px;font-size:9px;margin:2px 3px 2px 0;}';
        html += '.mtSearch .tags{padding:6px 14px 10px;border-top:1px
solid #cce0ee;}';
        html += '.mtSearch .ld{display:none;}';
        html += '@media
print{body{background:#fff;padding:20px;}.export-header{-webkit-print-color-adju
        html += TC + 'style' + G + TC + 'head' + G + T + 'body' + G;
        html += T + 'div class="export-header"' + G;
        html += T + 'h1' + G + '\uD83D\uDD0D HELIKON Chat Export' + TC +
'h1' + G;
        html += T + 'div class="meta"' + G + dateStr + ' ' + timeStr + '
\u2022 docs.trialanderror.tech' + TC + 'div' + G;
        html += TC + 'div' + G;
        var allNodes = log.childNodes;
        var ni;
        for (ni = 0; ni < allNodes.length; ni++) {
            var node = allNodes[ni];
            if (node.nodeType !== 1) continue;
            var cls = node.className || '';
            if (cls.indexOf('mtSearch') !== -1) { html += node.outerHTML
|| (T + 'div class="mtSearch"' + G + node.innerHTML + TC + 'div' + G);
continue; }
            if (cls.indexOf('typing') !== -1) continue;
            if (cls.indexOf('dual-chain') !== -1) { html += T + 'div
class="msg msg-dual"' + G + (node.textContent || '') + TC + 'div' + G;
continue; }
            if (cls.indexOf('fact') !== -1) { html += T + 'div class="msg
msg-fact"' + G + (node.textContent || '') + TC + 'div' + G; continue; }
            if (cls.indexOf('system') !== -1) { html += T + 'div
class="msg msg-system"' + G + (node.textContent || '') + TC + 'div' + G;
```

```
continue; }
            if (cls.indexOf('user') !== -1) { html += T + 'div class="msg
msg-user"' + G + (node.textContent || node.innerText || '') + TC + 'div'
+ G; continue; }
            if (cls.indexOf('buddy') !== -1) {
                var bCls = 'msg msg-buddy';
                if (cls.indexOf('claude') !== -1) bCls += ' claude';
                html += T + 'div class="' + bCls + '"' + G +
node.innerHTML + TC + 'div' + G;
                continue;
            }
        }
        html += TC + 'body' + G + TC + 'html' + G;
        var printWin = window.open('', '_blank');
        printWin.document.write(html);
        printWin.document.close();
        printWin.onload = function() { printWin.print(); };
        showBubble('PDF wird erstellt...', 3000);
    }
    if (exportBtn) exportBtn.addEventListener('click', exportPdf);

    /* === SCORE === */
    function calcScore() {
        if (ss.reqs === 0) { var r0 = new Object(); r0.score = 0;
r0.grade = 'Noch keine Daten'; r0.color = '#999'; return r0; }
        var avgMs = Math.round(ss.timeMs / ss.reqs);
        var avgTok = Math.round((ss.tokIn + ss.tokOut) / ss.reqs);
        var raw = 100;
        if (avgMs > 500) raw -= Math.min((avgMs - 500) / 50, 30);
        if (avgTok > 200) raw -= Math.min((avgTok - 200) / 20, 20);
        raw -= (ss.errors / ss.reqs) * 50;
        raw = Math.max(0, Math.min(100, Math.round(raw)));
        var grade = ''; var color = '#999';
        if (raw >= 90) { grade = '\u2B50 Prompt-Meister'; color =
'#4caf50'; }
        else if (raw >= 75) { grade = '\u26A1 Speed-Runner'; color =
'#8bc34a'; }
        else if (raw >= 60) { grade = '\u2705 Effizient'; color =
'#ffc107'; }
        else if (raw >= 40) { grade = '\u23F3 Gem\u00FCtlich'; color =
'#ff9800'; }
        else { grade = '\u2618 Energie-Verschwender'; color = '#f44336'; }
        var res = new Object(); res.score = raw; res.grade = grade;
res.color = color; return res;
    }

    function sr(label, value) { return tag('div', 'statRow', tag('span',
'statLabel', label) + tag('span', 'statValue', value)); }

    function updateStats() {
```

```
        if (!statsDiv) return;
        var elapsed = Math.floor((Date.now() - ss.start) / 1000);
        var mins = Math.floor(elapsed / 60); var secs = elapsed % 60;
        var totTok = ss.tokIn + ss.tokOut;
        var kwh = totTok * ENERGY_PER_TOKEN;
        var sc = calcScore();
        var ico = function(i) { return tagS('em', 'statIcon', '', i); };
        var h = tag('div', 'scoreBox', tag('span', 'scoreLabel',
'EFFIZIENZ-SCORE') + tagS('span', 'scoreNum', 'color:' + sc.color, '' +
sc.score) + tagS('span', 'scoreGrade', 'color:' + sc.color, sc.grade));
        h += tag('div', 'statsTitle', ico('\u2699') + 'Session');
        h += sr(ico('\u23F1') + 'Dauer', mins + 'm ' + secs + 's');
        h += sr(ico('\u2709') + 'Total Anfragen', '' + ss.reqs);
        h += sr(ico('\u26A0') + 'Fehler', '' + ss.errors);
        h += sr(ico('\u2211') + 'Tokens gesamt', '' + totTok);
        if (providers.p1.ready) {
            var avgP1 = ssP1.reqs > 0 ? Math.round(ssP1.timeMs /
ssP1.reqs) : 0;
            h += T + 'div class="statSection"' + G;
            h += tag('div', 'statsTitle', ico('\u2699') +
providers.p1.name + tagS('span', 'statProvName local', '',
providers.p1.model));
            h += sr(ico('\u2197') + 'Gesendet', '' + ssP1.reqs);
            h += sr(ico('\u26A0') + 'Fehler', '' + ssP1.errors);
            h += sr(ico('\u23F3') + 'Avg Latenz', avgP1 + ' ms');
            h += sr(ico('\u25B2') + 'Tokens rein', '' + ssP1.tokIn);
            h += sr(ico('\u25BC') + 'Tokens raus', '' + ssP1.tokOut);
            h += TC + 'div' + G;
        }
        if (providers.p2.ready) {
            var avgP2 = ssP2.reqs > 0 ? Math.round(ssP2.timeMs /
ssP2.reqs) : 0;
            h += T + 'div class="statSection"' + G;
            h += tag('div', 'statsTitle', ico('\u26A1') +
providers.p2.name + tagS('span', 'statProvName claude', '',
providers.p2.model));
            h += sr(ico('\u2197') + 'Gesendet', '' + ssP2.reqs);
            h += sr(ico('\u26A0') + 'Fehler', '' + ssP2.errors);
            h += sr(ico('\u23F3') + 'Avg Latenz', avgP2 + ' ms');
            h += sr(ico('\u25B2') + 'Tokens rein', '' + ssP2.tokIn);
            h += sr(ico('\u25BC') + 'Tokens raus', '' + ssP2.tokOut);
            h += TC + 'div' + G;
        }
        h += tag('div', 'statsTitle', ico('\u2618') + 'Energie');
        h += sr(ico('\u26A1') + 'Verbrauch', kwh.toFixed(6) + ' kWh');
        var comp = '';
        if (kwh > 0) { var ledH = kwh / 0.01; if (ledH < 0.01) comp =
'weniger als 1 Sek LED'; else if (ledH < 1) comp = (ledH * 60).toFixed(0)
+ ' Min LED'; else comp = ledH.toFixed(1) + ' Std LED'; }
        else { comp = 'noch nix verbraucht'; }
```

```
        h += sr(ico('\u2600') + 'Vergleich', comp);
        statsDiv.innerHTML = h;
    }

    if (statsBtn) {
        statsBtn.addEventListener('click', function() {
            statsVisible = !statsVisible;
            if (statsVisible) { updateStats(); statsDiv.className =
'visible'; statsBtn.className = 'active'; showBubble('Nerd-Modus!',
3000); }
            else { statsDiv.className = ''; statsBtn.className = ''; }
        });
    }

    /* === v2.6: FULLSCREEN TOGGLE === */
    var isFullscreen = false;
    var bBody = document.getElementById('topic-body');
    if (fsBtn && buddyFrame) {
        fsBtn.addEventListener('click', function() {
            isFullscreen = !isFullscreen;
            if (isFullscreen) {
                buddyFrame.className = (buddyFrame.className || '') + '
fullscreen';
                if (bBody) bBody.className = (bBody.className || '') + '
helikon-fullscreen';
                fsBtn.className = 'active';
                fsBtn.innerHTML = '\u2715 Normal';
                showBubble('Vollbild!', 2000);
                buddyFrame.scrollTop = 0;
                log.scrollTop = log.scrollHeight;
            } else {
                buddyFrame.className = (buddyFrame.className ||
'').replace(' fullscreen', '').replace(/^\s+|\s+$/g, '');
                if (bBody) bBody.className = (bBody.className ||
'').replace(' helikon-fullscreen', '').replace(/^\s+|\s+$/g, '');
                fsBtn.className = '';
                fsBtn.innerHTML = '\u26F6 Vollbild';
                showBubble('Normalansicht', 2000);
            }
        });
        document.addEventListener('keydown', function(e) {
            if (e.keyCode === 27 && isFullscreen) {
                isFullscreen = false;
                buddyFrame.className = (buddyFrame.className ||
'').replace(' fullscreen', '').replace(/^\s+|\s+$/g, '');
                if (bBody) bBody.className = (bBody.className ||
'').replace(' helikon-fullscreen', '').replace(/^\s+|\s+$/g, '');
                fsBtn.className = '';
                fsBtn.innerHTML = '\u26F6 Vollbild';
            }
```

```
        });
    }

    /* === MINDTOUCH TRIBUTE === */
    if (avatar && mtOverlay) {
        avatar.addEventListener('click', function() {
            var h = '';
            h += tag('div', 'mtTitle', '\u2764\uFE0F MindTouch Core (2006
\u2013 2013)');
            h += tag('div', 'mtQuote', '"This is the End, Beautiful
Friend"');
            h += T + 'div class="mtSection"' + G;
            h += 'Am 9. April 2013 schrieb Aaron Fulkerson den letzten
Blogpost.';
            h += T + 'br /' + G + T + 'br /' + G;
            h += 'MindTouch Core und MindTouch Platform wurden
eingestellt. Was als Fork von MediaWiki begann (DekiWiki, 2005), wurde zu
einem der Top-5 Open-Source-Projekte weltweit.';
            h += TC + 'div' + G;
            h += T + 'div class="mtSection"' + G;
            h += tagS('div', 'mtTeamTitle', 'font-size:9px; color:#999;
text-align:center; margin-bottom:10px; letter-spacing:1px', '\u2728 Was
MindTouch konnte \u2014 und was wir daraus machen');
            h += tag('div', 'mtFeatureItem', 'XHTML statt Wikitext,
WYSIWYG-Editing direkt im Browser');
            h += tag('div', 'mtFeatureItem', 'DekiScript: Eigene
Skriptsprache direkt in Wiki-Templates');
            h += tag('div', 'mtFeatureItem', 'REST-API auf C#/Mono \u2014
ihrer Zeit voraus');
            h += tag('div', 'mtFeatureItem', 'Lucene-Suchindex,
Template-Vererbung, Property-System');
            h += tag('div', 'mtFeatureItem', 'Top 5 Open-Source-Projekt
bei SourceForge (2007-2009)');
            h += tag('div', 'mtFeatureItem', 'Und heute: KI-Chat,
Dual-LLM, Voice-Synthese \u2014 alles auf der gleichen Pfanne');
            h += TC + 'div' + G;
            h += T + 'div class="mtSection"' + G;
            h += 'Danke an Aaron Fulkerson, Steve Bjorg und alle
Community-Contributors.';
            h += T + 'br /' + G + T + 'br /' + G;
            h += T + 'em' + G + 'Diese Seite l\u00E4uft noch immer auf
MindTouch Core 2013.' + TC + 'em' + G;
            h += T + 'br /' + G;
            h += T + 'em' + G + 'Manche Software stirbt nie wirklich.' +
TC + 'em' + G;
            h += T + 'br /' + G + T + 'br /' + G;
            h += T + 'strong' + G + 'Auf alten Pfannen lernt man kochen.'
+ TC + 'strong' + G;
            h += TC + 'div' + G;
            h += T + 'div id="mtTeam"' + G;
```

```
            h += tagS('div', 'mtTeamTitle', 'font-size:10px;
margin-bottom:14px', '\u{1F91D} Das Team hinter dem Wahnsinn');
            if (teamImgUrl) {
                h += T + 'div style="text-align:center; margin:12px 0
16px 0"' + G;
                h += T + 'img id="mtTeamImg" alt="HELIKON TEAM"
style="max-width:100%; border:2px solid #f0d0a0; border-radius:8px;
image-rendering:pixelated" /' + G;
                h += TC + 'div' + G;
            }
            h += T + 'div class="mtCredit"' + G;
            h += tagS('div', 'mtCreditIcon sparkleIcon', '', '\u{1F373}');
            h += T + 'div' + G;
            h += tag('span', 'mtCreditName', 'MindTouch Core');
            h += tag('span', 'mtCreditRole', 'Die alte Pfanne \u2014
DekiWiki/MindTouch Core 2013');
            h += tag('span', 'mtCreditHint', 'Das Fundament. Debian
Squeeze, PHP 5.3, Mono, Lucene. L\u00E4uft und l\u00E4uft.');
            h += TC + 'div' + G + TC + 'div' + G;
            h += T + 'div class="mtCredit"' + G;
            h += tagS('div', 'mtCreditIcon rainbowIcon', '',
'\u{1F468}\u200D\u{1F4BB}');
            h += T + 'div' + G;
            h += tag('span', 'mtCreditName', 'David / Minocula');
            h += tag('span', 'mtCreditRole', 'IMS with AI Chat \u2014
Visi\u00F6nar, Koch, Regenbogen-Blick');
            h += tag('span', 'mtCreditHint', '"Ich hab da was gesehen..."
\u2014 Sieht Dinge, die andere nicht sehen. Macht daraus Systeme.');
            h += TC + 'div' + G + TC + 'div' + G;
            h += T + 'div class="mtCredit"' + G;
            h += tagS('div', 'mtCreditIcon sparkleIcon', '', '\u2728');
            h += T + 'div' + G;
            h += tag('span', 'mtCreditName', 'Claude / Anthropic');
            h += tag('span', 'mtCreditRole', 'KI-Partner \u2014 Code,
Templates, Ideen, Debugging');
            h += tag('span', 'mtCreditHint', 'DekiScript-kompatibel seit
Tag 1. Schreibt new Object() statt {} ohne zu murren.');
            h += TC + 'div' + G + TC + 'div' + G;
            h += tagS('div', '', 'text-align:center; margin-top:16px;
font-size:8px; color:#bb8833; letter-spacing:1px', '\u{1F5A5}\uFE0F
docs.trialanderror.tech \u2014 Auf alten Pfannen lernt man kochen.');
            h += TC + 'div' + G;
            mtContent.innerHTML = h;
            if (teamImgUrl) { var tImg =
document.getElementById('mtTeamImg'); if (tImg) tImg.src = teamImgUrl; }
            mtOverlay.style.display = 'flex';
            setGif('happy'); showBubble('Danke, MindTouch!', 5000);
            setTimeout(function() { setGif('idle'); }, 5000);
        });
    }
```

```
    if (mtClose) { mtClose.addEventListener('click', function() {
mtOverlay.style.display = 'none'; }); }
    if (mtOverlay) { mtOverlay.addEventListener('click', function(e) { if
(e.target === mtOverlay) mtOverlay.style.display = 'none'; }); }

    /* === HANDLE REPLY === */
    function handleReply(reply, providerName, providerType) {
        history.push(makeMsg('assistant', reply));
        /* v2.6.6: Immer addProviderMsg() nutzen (mit .msgProvider-Badge),
         * auch im Single Mode — nicht nur im Dual Mode. Grund: das Voice-
         * Template identifiziert den Provider ueber den Badge-Text
(echter
         * Name), nicht ueber data-provider (nur 'claude'/'local' = Typ,
         * nicht Identitaet — bricht sobald beide Provider denselben Typ
         * haben, z.B. beide 'anthropic'). */
        var msgDiv = addProviderMsg(reply, providerName, providerType);
        var logData = new Object();
        logData.provider = providerName; logData.model = ss.model;
        logData.tokens = ss.tokOut; logData.latency = ss.lastMs;
logData.message = reply;
        logToServer('bot_reply', logData);
        if (reply.length <= SHORT_LEN && !dualMode) { showBubble(reply,
6000); }
        else {
            var dot = reply.indexOf('. ');
            var summary;
            if (dot > 0 && dot < 80) summary = reply.substring(0, dot +
1);
            else if (reply.length > 60) summary = reply.substring(0, 57)
+ '...';
            else summary = reply;
            showBubble(dualMode ? providerName + ': OK' : summary, 6000);
        }
        setGif('happy');
        setTimeout(function() { setGif('idle'); }, 3000);
        if (statsVisible) updateStats();
    }

    /* === LLM CALLS === */
    function callOpenAI(prov, searchContext, callback) {
        var fullSys = prov.sysMsg || '';
        if (prov.staticInfo) fullSys += '\n\n' + prov.staticInfo;
        if (searchContext) fullSys += searchContext;
        if (window.helikonCallMode) fullSys += '\n\n[CALL MODE: Antworte
kurz und praegnant, max 2-3 Saetze.]';
        if (dualChaining) fullSys += '\n\n[DUAL-GESPRAECH: Du diskutierst
mit einem anderen KI-Modell. Antworte auf den vorherigen Beitrag, bringe
eigene Perspektiven ein, stelle Fragen oder widersprich konstruktiv.
Nutze ' + AT + (prov === providers.p1 ? providers.p2.name.toLowerCase() :
providers.p1.name.toLowerCase()) + ' wenn du das andere Modell direkt
```

```
ansprechen willst. Halte deine Antwort KURZ: max 2-3 Saetze, nur das
Wichtigste. Diskutiere praegnant.]';
        var messages = []; messages.push(makeMsg('system', fullSys));
        var i; for (i = 0; i < history.length; i++)
messages.push(history[i]);
        var bodyObj = new Object(); bodyObj.model = prov.model;
bodyObj.messages = messages; bodyObj.max_tokens = 1000;
        var xhr = new XMLHttpRequest();
        xhr.open('POST', prov.url, true);
        xhr.setRequestHeader('Content-Type', 'application/json');
        if (prov.key) xhr.setRequestHeader('Authorization', 'Bearer ' +
prov.key);
        xhr.onload = function() {
            if (xhr.status === 200) { try { var data =
JSON.parse(xhr.responseText); callback(null,
data.choices[0].message.content, data); } catch (e) {
callback('Parse-Fehler'); } }
            else { callback('API-Fehler ' + xhr.status); }
        };
        xhr.onerror = function() { callback('Nicht erreichbar'); };
        currentXhr = xhr;
        xhr.send(JSON.stringify(bodyObj));
    }

    function callAnthropic(prov, searchContext, callback) {
        var fullSys = prov.sysMsg || '';
        if (prov.staticInfo) fullSys += '\n\n' + prov.staticInfo;
        if (searchContext) fullSys += searchContext;
        if (window.helikonCallMode) fullSys += '\n\n[CALL MODE: Antworte
kurz und praegnant, max 2-3 Saetze.]';
        if (dualChaining) fullSys += '\n\n[DUAL-GESPRAECH: Du diskutierst
mit einem anderen KI-Modell. Antworte auf den vorherigen Beitrag, bringe
eigene Perspektiven ein, stelle Fragen oder widersprich konstruktiv.
Nutze ' + AT + (prov === providers.p2 ? providers.p1.name.toLowerCase() :
providers.p2.name.toLowerCase()) + ' wenn du das andere Modell direkt
ansprechen willst. Halte deine Antwort KURZ: max 2-3 Saetze, nur das
Wichtigste. Diskutiere praegnant.]';
        var msgs = []; var i;
        for (i = 0; i < history.length; i++) { if (history[i].role !==
'system') msgs.push(makeMsg(history[i].role, history[i].content)); }
        var bodyObj = new Object(); bodyObj.model = prov.model;
bodyObj.max_tokens = 1000; bodyObj.system = fullSys; bodyObj.messages =
msgs;
        var xhr = new XMLHttpRequest();
        xhr.open('POST', prov.url, true);
        xhr.setRequestHeader('Content-Type', 'application/json');
        xhr.setRequestHeader('x-api-key', prov.key);
        xhr.setRequestHeader('anthropic-version', '2023-06-01');
        xhr.setRequestHeader('anthropic-dangerous-direct-browser-access',
'true');
```

```
        xhr.onload = function() {
            if (xhr.status === 200) {
                try {
                    var data = JSON.parse(xhr.responseText);
                    var reply = ''; var ci;
                    for (ci = 0; ci < data.content.length; ci++) { if
(data.content[ci].type === 'text') reply += data.content[ci].text; }
                    callback(null, reply, data);
                } catch (e) { callback('Parse-Fehler'); }
            } else { callback('API-Fehler ' + xhr.status); }
        };
        xhr.onerror = function() { callback('Nicht erreichbar'); };
        currentXhr = xhr;
        xhr.send(JSON.stringify(bodyObj));
    }

    function callProvider(providerKey, searchContext, callback) {
        var prov = providers[providerKey];
        if (!prov || !prov.ready) { callback('Provider nicht
konfiguriert'); return; }
        var startTime = Date.now();
        ss.provider = prov.name; ss.model = prov.model;
        var psub = (providerKey === 'p1') ? ssP1 : ssP2;
        var wrapCb = function(err, reply, data) {
            var latency = Date.now() - startTime;
            ss.lastMs = latency; ss.timeMs += latency; ss.reqs++;
            psub.lastMs = latency; psub.timeMs += latency; psub.reqs++;
            if (err) { ss.errors++; psub.errors++; updateLeds();
callback(err); }
            else {
                ss.chOut += reply.length;
                if (data && data.usage) {
                    var tIn = data.usage.prompt_tokens ||
data.usage.input_tokens || 0;
                    var tOut = data.usage.completion_tokens ||
data.usage.output_tokens || 0;
                    ss.tokIn += tIn; ss.tokOut += tOut; psub.tokIn +=
tIn; psub.tokOut += tOut;
                } else {
                    var est = estimateTokens(reply);
                    ss.tokIn += est; ss.tokOut += est; psub.tokIn += est;
psub.tokOut += est;
                }
                updateLeds();
                callback(null, reply);
            }
        };
        if (prov.type === 'anthropic') callAnthropic(prov, searchContext,
wrapCb);
        else callOpenAI(prov, searchContext, wrapCb);
```

```
    }

    /* === DUAL HARD ABORT === */
    function abortDualChain() {
        dualAbortFlag = true; dualChaining = false;
        if (currentXhr) { try { currentXhr.abort(); } catch(e) {}
currentXhr = null; }
        busy = false; btn.disabled = false; input.focus();
        addMsg('\u{1F6D1} Dual-Kette abgebrochen', 'dual-chain');
        setStatus('Dual Mode gestoppt', 'warn');
        showBubble('Abgebrochen!', 3000); setGif('idle');
        logToServer('dual_abort', new Object());
    }

    /* === DUAL CHAIN === */
    var dualSearchCtx = '';

    function dualChainNext(reply, fromProviderKey) {
        if (dualAbortFlag) { dualAbortFlag = false; return; }
        var targetKey = (fromProviderKey === 'p1') ? 'p2' : 'p1';
        if (dualRound >= DUAL_MAX_ROUNDS) {
            if (dualAbortFlag) { dualAbortFlag = false; return; }
            addMsg('\u{1F6D1} Dual-Kette: ' + dualRound + ' Runden
abgeschlossen', 'dual-chain');
            if (dualSummaryEnabled) {
                setStatus('Zusammenfassung wird erstellt...', 'warn');
                showBubble('Fasse zusammen...', 5000);
                dualGenerateSummary();
            } else {
                dualChaining = false; busy = false; btn.disabled = false;
input.focus();
                setStatus('Dual Mode | bereit (' + dualRound + '
Runden)', 'ok');
                showBubble('Fertig!', 3000); setGif('happy');
                setTimeout(function() { setGif('idle'); }, 3000);
            }
            return;
        }
        if (!providers[targetKey].ready) {
            addMsg('\u26A0 ' + providers[targetKey].name + ' nicht
verfuegbar', 'system');
            dualChaining = false; busy = false; btn.disabled = false;
return;
        }
        dualRound++;
        var fromName = providers[fromProviderKey].name;
        var targetName = providers[targetKey].name;
        addMsg('\u{1F517} ' + fromName + ' \u2192 ' + targetName + '
(Runde ' + dualRound + '/' + DUAL_MAX_ROUNDS + ')', 'dual-chain');
        setStatus(targetName + ' antwortet... (Runde ' + dualRound + ')',
```

```
'warn');
        history.push(makeMsg('user', '[' + fromName + ']: ' + reply));
        var typing = addMsg(targetName + ' denkt nach...', 'buddy
typing');
        callProvider(targetKey, dualSearchCtx, function(err, nextReply) {
            if (typing.parentNode) log.removeChild(typing);
            if (dualAbortFlag) { dualAbortFlag = false; return; }
            if (err) {
                addMsg('[' + targetName + '] Fehler: ' + err, 'system');
                dualChaining = false; busy = false; btn.disabled = false;
            } else {
                handleReply(nextReply, targetName,
providers[targetKey].type);
                setTimeout(function() { dualChainNext(nextReply,
targetKey); }, 500);
            }
        });
    }

    /* === DUAL SUMMARY === */
    function dualGenerateSummary() {
        if (dualAbortFlag) { dualAbortFlag = false; dualChaining = false;
busy = false; btn.disabled = false; return; }
        var summaryPrompt = '[ZUSAMMENFASSUNG] Das Dual-Gespraech ist
nach ' + dualRound + ' Runden beendet. Fasse die wichtigsten
Erkenntnisse, Entscheidungen und offenen Punkte zusammen. Erstelle eine
klare Liste:\n1. Was wurde besprochen\n2. Was wurde entschieden\n3. Was
ist noch offen / naechste Schritte\nSei praegnant und strukturiert.';
        /* v2.6.5: Nur EIN Provider erstellt die Zusammenfassung (Model 1,
         * mit Fallback auf Model 2 falls Model 1 zwischenzeitlich nicht
         * mehr ready sein sollte). */
        var summarizerKey = providers.p1.ready ? 'p1' :
(providers.p2.ready ? 'p2' : null);
        if (!summarizerKey) { dualChaining = false; busy = false;
btn.disabled = false; return; }
        var prov = providers[summarizerKey];
        addMsg('\u{1F4CB} Zusammenfassung wird erstellt...',
'dual-chain');
        history.push(makeMsg('user', summaryPrompt));
        callProvider(summarizerKey, '', function(err, reply) {
            history.pop();
            if (dualAbortFlag) { dualAbortFlag = false; dualChaining =
false; busy = false; btn.disabled = false; return; }
            if (err) { addMsg('[' + prov.name + '] Zusammenfassung
fehlgeschlagen: ' + err, 'system'); }
            else { addMsg('\u{1F4CB} ' + prov.name + ' \u2014
Zusammenfassung:', 'dual-chain'); addProviderMsg(reply, prov.name,
prov.type); }
            dualChaining = false; busy = false; btn.disabled = false;
input.focus();
```

```
            setStatus('Dual Mode | bereit (' + dualRound + ' Runden)',
'ok');
            showBubble('Fertig!', 3000); setGif('happy');
            setTimeout(function() { setGif('idle'); }, 4000);
        });
    }

    /* ============================================================
     * SEND — v2.6.1: logToServer user_msg mit message-Inhalt
     *        v2.6.8: Suche via /site/query, Prompt-Bereinigung,
     *                kein isShortQuery-Gate mehr
     * ============================================================ */
    function send() {
        var msg = input.value.replace(/^\s+|\s+$/g, '');
        if (!msg || busy) return;
        if (!providers.p1.ready && !providers.p2.ready) {
            addMsg('Kein LLM konfiguriert.', 'system'); showBubble('Keine
API!', 3000);
            setGif('error'); setStatus('Nicht konfiguriert', 'err');
            setTimeout(function() { setGif('idle'); }, 4000); return;
        }
        addMsg(msg, 'user'); input.value = '';
        history.push(makeMsg('user', msg));
        ss.chIn += msg.length;
        /* v2.6.1 FIX: message-Inhalt korrekt mitgeben */
        var logMsgData = new Object();
        logMsgData.message = msg;
        logToServer('user_msg', logMsgData);
        busy = true; btn.disabled = true;
        setGif('loading'); showBubble('...', 30000);

        var mentionTarget = detectMention(msg);

        var doCall = function(searchCtx) {
            dualSearchCtx = searchCtx;
            if (dualMode) {
                dualRound = 1; dualChaining = true; dualAbortFlag = false;
                var startKey = mentionTarget || 'p1';
                if (!providers[startKey].ready) startKey = (startKey ===
'p1') ? 'p2' : 'p1';
                var startName = providers[startKey].name;
                setStatus(startName + ' antwortet... (Runde 1/' +
DUAL_MAX_ROUNDS + ')', 'warn');
                var typing = addMsg(startName + ' denkt nach...', 'buddy
typing');
                callProvider(startKey, searchCtx, function(err, reply) {
                    if (typing.parentNode) log.removeChild(typing);
                    if (dualAbortFlag) { dualAbortFlag = false; return; }
                    if (err) {
                        addMsg('[' + startName + '] Fehler: ' + err,
```

```
'system');
                        dualChaining = false; busy = false; btn.disabled
= false;
                    } else {
                        handleReply(reply, startName,
providers[startKey].type);
                        setTimeout(function() { dualChainNext(reply,
startKey); }, 500);
                    }
                });
            } else {
                var provKey = mentionTarget || activeProvider;
                if (!providers[provKey].ready) provKey = activeProvider;
                var typing = addMsg('...', 'buddy typing');
                callProvider(provKey, searchCtx, function(err, reply) {
                    if (typing.parentNode) log.removeChild(typing);
                    if (err) {
                        addMsg('Fehler: ' + err, 'system');
showBubble('Fehler!', 3000);
                        setGif('error'); setTimeout(function() {
setGif('idle'); }, 4000);
                        setStatus(err, 'err'); updateLeds();
                    } else {
                        handleReply(reply, providers[provKey].name,
providers[provKey].type);
                        setStatus(providers[provKey].name + ' | bereit',
'ok');
                    }
                    busy = false; btn.disabled = false; input.focus();
                });
            }
        };

        /* --- firnCore customization --- v2.6.8 */
        var searchQ = cleanSearchQuery(stripMention(msg));
        if (isGreeting(msg) || searchQ.length < 2) {
            sdbg('keine Suche (Gruss oder nur Fuellwoerter): "' + msg +
'"');
            setStatus('antworte...', 'warn');
            doCall('');
        } else {
            var sWords = searchWords(searchQ);
            sdbg('Prompt: "' + msg + '" / Query: "' + searchQ + '"');
            setStatus('durchsuche Doku...', 'warn');
            searchWiki(searchQ, sWords, function(results, files) {
                var hits = results.length + files.length;
                if (hits === 0) {
                    setStatus('denkt nach...', 'warn');
                    doCall('');
                } else {
```

```
                    setStatus('laedt Seiteninhalte... (' + hits + '
Treffer)', 'warn');
                    buildSearchContext(results, files, sWords,
function(ctx) {
                        setStatus('denkt nach... (' + hits + ' Treffer)',
'warn');
                        doCall(ctx);
                    });
                }
            });
        }
    }

    /* === LLM SWITCH === */
    function updateSwitchUI() {
        llmBtn1.className = 'llmBtn'; llmBtn2.className = 'llmBtn';
llmBtnDual.className = 'llmBtn';
        llmBtn1.disabled = !providers.p1.ready;
        llmBtn2.disabled = !providers.p2.ready;
        llmBtnDual.disabled = (!providers.p1.ready ||
!providers.p2.ready);
        if (dualMode) llmBtnDual.className = 'llmBtn active-dual';
        else if (activeProvider === 'p1') llmBtn1.className = 'llmBtn
active';
        else llmBtn2.className = 'llmBtn active';
        if (!dualMode && providers[activeProvider])
setStatus(providers[activeProvider].name + ' | bereit', 'ok');
        else if (dualMode) setStatus('Dual Mode | bereit', 'ok');
        setLed(llmBtn1, providers.p1.ready ? 'ok' : '');
        setLed(llmBtn2, providers.p2.ready ? 'ok' : '');
        updateLeds();
        if (dualInfo) {
            if (dualMode) {
                if (dualInfoText) dualInfoText.innerHTML = '\u{1F517}
DUAL MODE: Nutze ' + AT + providers.p1.name.toLowerCase().split(' ')[0] +
' oder ' + AT + providers.p2.name.toLowerCase().split(' ')[0] + ' um
gezielt zu starten. Die KIs wechseln sich ab und fassen am Ende
zusammen.';
                dualInfo.className = 'visible';
                if (dualRndVal) dualRndVal.textContent = '' +
DUAL_MAX_ROUNDS;
            } else { dualInfo.className = ''; }
        }
    }

    llmBtn1.addEventListener('click', function() { if
(!providers.p1.ready) return; dualMode = false; activeProvider = 'p1';
updateSwitchUI(); showBubble(providers.p1.name + ' aktiv', 2000);
logToServer('provider_switch', new Object()); });
    llmBtn2.addEventListener('click', function() { if
```

```
(!providers.p2.ready) return; dualMode = false; activeProvider = 'p2';
updateSwitchUI(); showBubble(providers.p2.name + ' aktiv', 2000);
logToServer('provider_switch', new Object()); });
    llmBtnDual.addEventListener('click', function() {
        if (!providers.p1.ready || !providers.p2.ready) return;
        if (dualMode && dualChaining) { abortDualChain(); dualMode =
false; updateSwitchUI(); return; }
        dualMode = !dualMode;
        updateSwitchUI();
        showBubble(dualMode ? 'Dual-Modus aktiviert!' :
providers[activeProvider].name + ' aktiv', 3000);
        logToServer('provider_switch', new Object());
    });

    if (dualRndDown) dualRndDown.addEventListener('click', function() {
if (DUAL_MAX_ROUNDS > DUAL_MIN_ROUNDS) { DUAL_MAX_ROUNDS--;
dualRndVal.textContent = '' + DUAL_MAX_ROUNDS; } });
    if (dualRndUp) dualRndUp.addEventListener('click', function() { if
(DUAL_MAX_ROUNDS < DUAL_CEIL_ROUNDS) { DUAL_MAX_ROUNDS++;
dualRndVal.textContent = '' + DUAL_MAX_ROUNDS; } });

    /* v2.6.7: Zusammenfassung-Checkbox — Standard AUS (dualSummaryEnabled
     * ist bereits false initialisiert). Checkbox startet ungecheckt, da
     * HTML kein "checked"-Attribut hat. */
    if (dualSummaryCheck) {
        dualSummaryCheck.checked = false;
        dualSummaryCheck.addEventListener('change', function() {
            dualSummaryEnabled = !!dualSummaryCheck.checked;
            showBubble(dualSummaryEnabled ? 'Zusammenfassung: an' :
'Zusammenfassung: aus', 2000);
        });
    }

    btn.addEventListener('click', send);
    input.addEventListener('keydown', function(e) { if (e.keyCode === 13)
send(); });

    /* === INIT === */
    setGif('loading'); setStatus('lade konfiguration...'); input.disabled
= true; btn.disabled = true;

    fetchAllProperties(function(props) {
        providers.p1.url = props['buddy.apiurl1'] || '';
        providers.p1.key = props['buddy.apikey1'] || '';
        if (props['buddy.model1']) providers.p1.model =
props['buddy.model1'];
        if (props['buddy.name1']) providers.p1.name =
props['buddy.name1'];
        if (props['buddy.type1']) providers.p1.type =
props['buddy.type1'];
```

```
        providers.p1.ready = !!providers.p1.url;

        providers.p2.url = props['buddy.apiurl2'] || '';
        providers.p2.key = props['buddy.apikey2'] || '';
        if (props['buddy.model2']) providers.p2.model =
props['buddy.model2'];
        if (props['buddy.name2']) providers.p2.name =
props['buddy.name2'];
        if (props['buddy.type2']) providers.p2.type =
props['buddy.type2'];
        providers.p2.ready = !!providers.p2.url;

        if (providers.p1.name) { var n1 =
providers.p1.name.toLowerCase().split(' ')[0]; if
(providers.p1.mentions.indexOf(n1) === -1)
providers.p1.mentions.push(n1); }
        if (providers.p2.name) { var n2 =
providers.p2.name.toLowerCase().split(' ')[0]; if
(providers.p2.mentions.indexOf(n2) === -1)
providers.p2.mentions.push(n2); }

        mtConfig.base = props['buddy.mtapi'] || '/@api/deki';
        mtConfig.key = props['buddy.mtapikey'] || '';
        mtConfig.ready = true;
        var teamImgFile = props['buddy.teamimg'] || '';
        if (teamImgFile) teamImgUrl = teamImgFile;

        if (llmBtn1) { var led1 = llmBtn1.querySelector('.llmLed');
llmBtn1.textContent = ' ' + providers.p1.name; if (led1)
llmBtn1.insertBefore(led1, llmBtn1.firstChild); }
        if (llmBtn2) { var led2 = llmBtn2.querySelector('.llmLed');
llmBtn2.textContent = ' ' + providers.p2.name; if (led2)
llmBtn2.insertBefore(led2, llmBtn2.firstChild); }
        if (!providers.p1.ready && providers.p2.ready) activeProvider =
'p2';

        /* ============================================================
         * v2.6.4: PFLICHT — jeder konfigurierte Provider (apiurl gesetzt)
         * MUSS eigenen systemprompt + staticinfo haben. Kein Fallback auf
         * die alten geteilten buddy.systemprompt / buddy.staticinfo.
         * ============================================================ */
        providers.p1.sysPromptUrl = props['buddy.systemprompt1'] || '';
        providers.p1.staticInfoUrl = props['buddy.staticinfo1'] || '';
        providers.p2.sysPromptUrl = props['buddy.systemprompt2'] || '';
        providers.p2.staticInfoUrl = props['buddy.staticinfo2'] || '';

        function checkProviderPromptConfig(prov, idx) {
            var errs = [];
            if (!prov.url) return errs;
            if (!prov.sysPromptUrl) errs.push('buddy.systemprompt' + idx
```

```
+ ' fehlt');
            if (!prov.staticInfoUrl) errs.push('buddy.staticinfo' + idx +
' fehlt');
            return errs;
        }
        providers.p1.configErrors =
checkProviderPromptConfig(providers.p1, 1);
        providers.p2.configErrors =
checkProviderPromptConfig(providers.p2, 2);
        if (providers.p1.configErrors.length > 0) providers.p1.ready =
false;
        if (providers.p2.configErrors.length > 0) providers.p2.ready =
false;

        function ready() {
            var loaded = [];
            var ki;
            var keys = new Array('p1', 'p2');
            for (ki = 0; ki < keys.length; ki++) {
                var prov = providers[keys[ki]];
                if (!prov.url) continue;
                if (prov.configErrors.length > 0) {
                    loaded.push(prov.name + ': ' +
prov.configErrors.join(', '));
                } else {
                    loaded.push(prov.name + ': Prompt OK (' +
prov.sysMsg.length + ') | Static OK (' + prov.staticInfo.length + ')');
                }
            }
            setGif('idle'); input.disabled = false; btn.disabled = false;
            updateSwitchUI();
            var startProv = providers[activeProvider];
            if (startProv && startProv.ready)
logToServer('session_start', new Object());
            var anyConfigError = (providers.p1.configErrors.length > 0)
|| (providers.p2.configErrors.length > 0);
            if (providers.p1.ready || providers.p2.ready) {
                showBubble('Hallo! Wie kann ich helfen?', 5000);
                setStatus(loaded.join(' | '), anyConfigError ? 'warn' :
'');
                healthCheckProviders();
            } else if (anyConfigError) {
                showBubble('Prompt-Konfiguration fehlt!', 5000);
setGif('error');
                setStatus(loaded.join(' | '), 'err');
            } else { showBubble('Kein LLM konfiguriert!', 5000);
setGif('error'); setStatus('buddy.apiurl1 / buddy.apiurl2 fehlt', 'err');
}
            if (statsBtn) statsBtn.innerHTML = '\u2630 Stats';
            if (exportBtn) exportBtn.innerHTML = '\uD83D\uDCC4 PDF';
```

```
            if (fsBtn) fsBtn.innerHTML = '\u26F6 Vollbild';
        }

        function healthCheckProviders() {
            var keys = new Array('p1', 'p2');
            var ki;
            for (ki = 0; ki < keys.length; ki++) {
                (function(pk) {
                    var prov = providers[pk];
                    if (!prov.ready) return;
                    var btnEl = (pk === 'p1') ? llmBtn1 : llmBtn2;
                    var xhr = new XMLHttpRequest();
                    xhr.open('POST', prov.url, true);
                    xhr.setRequestHeader('Content-Type', 'application/
json');
                    xhr.timeout = 8000;
                    if (prov.type === 'anthropic') {
                        xhr.setRequestHeader('x-api-key', prov.key);
                        xhr.setRequestHeader('anthropic-version',
'2023-06-01');

xhr.setRequestHeader('anthropic-dangerous-direct-browser-access', 'true');
                    } else { if (prov.key)
xhr.setRequestHeader('Authorization', 'Bearer ' + prov.key); }
                    xhr.onload = function() { setLed(btnEl, 'ok'); };
                    xhr.onerror = function() { setLed(btnEl, 'err');
prov.ready = false; updateSwitchUI(); setStatus(prov.name + ': nicht
erreichbar!', 'err'); };
                    xhr.ontimeout = function() { setLed(btnEl, 'err');
prov.ready = false; updateSwitchUI(); setStatus(prov.name + ': Timeout!',
'err'); };
                    var testBody = new Object(); testBody.model =
prov.model; testBody.messages = [makeMsg('user', 'ping')];
testBody.max_tokens = 1;
                    if (prov.type === 'anthropic') testBody.system =
'reply ok';
                    xhr.send(JSON.stringify(testBody));
                })(keys[ki]);
            }
        }

        /* v2.6.4: Pro Provider parallel laden — nur wenn Config
vollstaendig
         * (sonst waere ein Fetch-Versuch auf eine fehlende URL sinnlos).
         * Ergebnis leer/nicht ladbar -> ebenfalls als Config-Fehler
werten. */
        function loadProviderPrompts(prov, idx, doneCb) {
            if (prov.configErrors.length > 0) { doneCb(); return; }
            var pending = 2;
            function finishOne() { pending--; if (pending <= 0) doneCb();
```

```
}
            fetchText(prov.sysPromptUrl, function(val) {
                if (!val) prov.configErrors.push('buddy.systemprompt' +
idx + ' nicht ladbar/leer');
                prov.sysMsg = val || '';
                finishOne();
            });
            fetchText(prov.staticInfoUrl, function(val) {
                if (!val) prov.configErrors.push('buddy.staticinfo' + idx
+ ' nicht ladbar/leer');
                prov.staticInfo = val || '';
                finishOne();
            });
        }

        var loadPending = 2;
        function loadDone() {
            loadPending--;
            if (loadPending <= 0) {
                if (providers.p1.configErrors.length > 0)
providers.p1.ready = false;
                if (providers.p2.configErrors.length > 0)
providers.p2.ready = false;
                ready();
            }
        }
        loadProviderPrompts(providers.p1, 1, loadDone);
        loadProviderPrompts(providers.p2, 2, loadDone);
    });

    /* === HEADER HOVER FACTS === */
    var headerFacts = new Object(); var hf = headerFacts;
    hf['Charles Babbage'] = new Array('Der Urvater aller Computer! Hat
die Analytical Engine nie fertig gebaut. Kennen wir das nicht
alle?','Babbage hasste Strassenmusikanten. H\u00E4tte er
Slack-Notifications erlebt...','Fun fact: Die britische Regierung gab ihm
17000 Pfund. Das w\u00E4re heute \u00FCber 3 Millionen!');
    hf['Ada Lovelace'] = new Array('Die erste Programmiererin der Welt!
Und das ohne Stack Overflow.','Ada schrieb den ersten Algorithmus bevor
es Computer gab. Talk about early adopter!','Tochter von Lord Byron.
Poesie und Code liegen n\u00E4her als man denkt.');
    hf['Konrad Zuse'] = new Array('Z3: Der erste funktionierende Computer
der Welt. Made in Germany!','Zuse hat nebenbei auch die erste h\u00F6here
Programmiersprache erfunden: Plankalk\u00FCl.','Er baute den Z1 im
Wohnzimmer seiner Eltern. Der erste Nerd-Keller war ein Wohnzimmer.');
    hf['Howard Aiken'] = new Array('Harvard Mark I: 16 Meter lang, 5
Tonnen schwer. Und weniger Rechenpower als deine Smartwatch.','Aiken
sagte: Die Welt braucht vielleicht 5 Computer. Ups.','Der Mark I klang
wie eine N\u00E4hfabrik. Noise-Cancelling gabs noch nicht.');
    hf['Alan Turing'] = new Array('Hat den Enigma-Code geknackt und den
```

```
Krieg verk\u00FCrzt. Nicht schlecht f\u00FCr einen Mathematiker.','Der
Turing-Test: Kann eine Maschine denken? Frag mich nochmal in 10
Jahren.','Turing liebte Langstreckenl\u00E4ufe. Debug-Methode: Rennen bis
die L\u00F6sung kommt.');
    hf['John von Neumann'] = new Array('Die Von-Neumann-Architektur:
Jeder Computer den du je benutzt hast basiert darauf.','Er konnte ganze B\
u00FCcher auswendig zitieren. Menschliches RAM auf Maximum.','Von Neumann
rechnete schneller im Kopf als fr\u00FChere Computer. Kein Witz.');
    hf['Steve Jobs bis 1976'] = new Array('Apple wurde in einer Garage
gegr\u00FCndet. Jedes Startup braucht eine Origin-Story.','Jobs war
barfuss bei wichtigen Meetings. Der erste Casual Friday.','Er ging nach
Indien um sich selbst zu finden. Kam zur\u00FCck und erfand Apple.');
    hf['IBM PC 5150'] = new Array('Der IBM PC: 4.77 MHz, 16 KB RAM. Dein
Browser-Tab braucht heute mehr.','1981 kostete er 1565 Dollar.
Inflation-bereinigt ca. 5000 Dollar. F\u00FCr 16 KB RAM!','IBM dachte PCs
w\u00E4ren ein Nischenprodukt. Klassischer Management-Move.');
    hf['Apple II'] = new Array('Der Apple II: Erster Computer mit
Farbgrafik f\u00FCr zuhause. Revolution!','Wozniak hat den Apple II quasi
alleine designed. Jobs hat ihn verkauft.','VisiCalc auf dem Apple II war
die erste Killer-App. Excel-Urahn!');
    hf['Steve Jobs bis 1985'] = new Array('Von Apple gefeuert. Von der
eigenen Firma. Das muss man erstmal schaffen.','Jobs Reality Distortion
Field: Er konnte dich \u00FCberzeugen dass schwarz weiss ist.','Nach
Apple gr\u00FCndete er NeXT. Spoiler: Er kommt zur\u00FCck.');
    hf['Diskette'] = new Array('Speichern-Icon bis heute! Kinder fragen:
Was ist dieses 3D-gedruckte Save-Symbol?','1.44 MB. Da passt nicht mal
ein Handyfoto drauf.','Das befriedigende Klick-Ger\u00E4usch beim
Einschieben. ASMR der 90er.');
    hf['Game Boy'] = new Array('Tetris auf dem Game Boy: Produktivit\
u00E4tskiller seit 1989.','Der Game Boy \u00FCberlebte einen
Bombenangriff im Golfkrieg. Nokia-Level H\u00E4rte.','118 Millionen St\
u00FCck verkauft. Und alle spielten Tetris.');
    hf['NeXTstation'] = new Array('Tim Berners-Lee erfand das Web auf
einem NeXT. Danke, Steve!','Der NeXT-W\u00FCrfel kostete 6500 Dollar.
Design hat seinen Preis.','NeXTSTEP wurde zu macOS. Die beste Rache ist
Erfolg.');
    hf['CD-ROM'] = new Array('700 MB! Das war mal unvorstellbar viel
Speicher.','Bitte nicht die gl\u00E4nzende Seite anfassen.
BITTE!','AOL-CDs als Untersetzer. Recycling der 90er.');
    hf['Netscape Navigator'] = new Array('Netscape: Der Browser der das
Internet popul\u00E4r machte. RIP.','Der Browserkrieg hat begonnen!
Netscape vs. Microsoft, Runde 1.','Aus Netscape wurde Mozilla Firefox.
Der Phoenix aus der Asche!');
    hf['Internet Explorer'] = new Array('Internet Explorer: Nur genutzt
um einen anderen Browser runterzuladen.','IE6 hat eine ganze Generation
von Webentwicklern traumatisiert.','Ruhe in Frieden, IE. Du warst... ein
Browser. Irgendwie.');
    hf['Bill Gates'] = new Array('640 KB sollten f\u00FCr jeden reichen.
Legend\u00E4res Fehlzitat!','Von Harvard abgebrochen. Eltern hassen
diesen Trick.','Ctrl+Alt+Delete war ein Fehler, sagt er selbst.
```

```
Wenigstens ehrlich.');
    hf['DVD und Blu-ray'] = new Array('DVD: Rewind war gestern! Aber
Regionalcodes waren die neue Nerverei.','Blu-ray vs HD-DVD: Der letzte
grosse Formatkrieg.','Wer hat noch eine DVD-Sammlung im Regal? Sei
ehrlich.');
    hf['Geschichte von Linux'] = new Array('Linus Torvalds: "Ich mache
ein freies Betriebssystem. Wird wohl nichts Grosses."','Linux l\u00E4uft
auf Supercomputern, Handys, K\u00FChlschr\u00E4nken und dem
Mars-Rover!','BTW, I use Arch. Der Veganer-Spruch der IT.');
    hf['List of Linux distributions'] = new Array('Es gibt \u00FCber 600
Linux-Distros. F\u00FCr jedes Problem eine eigene Distribution.','Ubuntu,
Debian, Arch, Fedora, Mint... Die Qual der Wahl.','Jeder Linux-User hat
schon 3x die Distro gewechselt. Mindestens.');
    hf['Handheld'] = new Array('Von Handhelds zu Smartphones. Die
Evolution der Ablenkung.','Palm Pilot! Wer erinnert sich an
Graffiti-Schrifterkennung?','Handhelds: Als Taschencomputer noch eine
eigene Kategorie waren.');
    hf['Personal Digital Assistant'] = new Array('PDA: Personal Digital
Assistant. Heute sagt man einfach Handy.','Mit dem Stylus auf dem PDA
tippen. Das war Premium-UX!','Newton MessagePad: Apples erster Flop. Oder
seiner Zeit voraus?');
    hf['Google'] = new Array('Googles erster Server war in Lego-Steinen
eingebaut. Kein Witz!','Googles erstes B\u00FCro: Eine Garage. Schon
wieder eine Garage.','"Don\'t be evil" hiess es mal. Heute heisst es...
naja.');
    hf['Notebook'] = new Array('Vom 10-Kilo-Laptop zum Ultrabook. Unsere
Rucks\u00E4cke danken es.','Erster Laptop: Osborne 1, 1981. 11 Kilo.
Armtraining inklusive.','Laptop im Bett: Modernes Lagerfeuer f\u00FCr
digitale Nomaden.');
    hf['USB-Stick'] = new Array('USB: Braucht immer 3 Versuche zum
Einstecken. Immer.','USB-C: Endlich egal welche Seite! Nur 20 Jahre hat
es gedauert.','Wer hat nicht mindestens 5 USB-Sticks in der Schublade
liegen?');
    hf['Youtube'] = new Array('Erstes Youtube-Video: "Me at the zoo". 19
Sekunden die alles \u00E4nderten.','Youtube frisst mehr Bandbreite als
TV. Die Zukunft ist hier.','Katzenvideos: Der wahre Grund warum das
Internet erfunden wurde.');
    hf['Tim Berners-Lee'] = new Array('Hat das World Wide Web erfunden.
Und nie Milliarden damit verdient. Respekt!','Erster Webserver lief auf
einem NeXT-W\u00FCrfel. Mit einem Zettel: "Nicht ausschalten!"','Ohne ihn
w\u00E4rst du jetzt nicht hier. Danke, Tim!');
    hf['Containervirtualisierung'] = new Array('Docker: Funktioniert auf
meinem Computer! Endlich auch auf deinem.','Container sind wie
Schiffscontainer: Alles drin, \u00FCberall einsetzbar.','kubectl get
pods: Der Befehl der DevOps-Engineers nicht schlafen l\u00E4sst.');
    hf['Smartphone OS'] = new Array('Android vs iOS: Der ewige
Glaubenskrieg. Schlimmer als Tabs vs Spaces.','Es gab mal Windows Phone.
Moment der Stille bitte.','Dein Smartphone hat mehr Power als die
Apollo-11-Rakete. Und du spielst Candy Crush.');
    hf['SaaS'] = new Array('Software as a Service: Fr\u00FCher kaufte man
```

```
Software. Heute mietet man sie. F\u00FCr immer.','SaaS: Wo jede App ein
monatliches Abo ist. Dein Konto weint leise.','Die Cloud ist nur der
Computer von jemand anderem. Romantisch, oder?');
    hf['Cloud Computing'] = new Array('Die Cloud: Klingt fluffy, ist ein
riesiges Rechenzentrum irgendwo in Irland.','Cloud Computing: Weil
"anderer-Leute-Computer" nicht so cool klingt.','Fr\u00FCher hatte man
Server im Keller. Jetzt hat Amazon sie. Fortschritt?');

    var factBubbles = new Array('Schon gewusst?', 'Fun Fact!', 'Wusstest
du?', 'Ach ja...', 'Apropos!', 'Hey, check das!');
    function pick(arr) { return arr[Math.floor(Math.random() *
arr.length)]; }
    var factCooldown = false;
    var areas = document.querySelectorAll('area[title]');
    var ai;
    for (ai = 0; ai < areas.length; ai++) {
        (function(area) {
            area.addEventListener('mouseenter', function() {
                if (factCooldown || busy) return;
                var title = area.getAttribute('title') || '';
                var facts = headerFacts[title];
                if (!facts) return;
                factCooldown = true;
                addMsg(title + ': ' + pick(facts), 'fact');
                setGif('happy'); showBubble(pick(factBubbles), 3000);
                setTimeout(function() { setGif('idle'); }, 3000);
                setTimeout(function() { factCooldown = false; }, 5000);
            });
        })(areas[ai]);
    }

})();
```