

Vollständiger Skript-Code: Google Ads MCC Bidding-Anomalie-Detektor (v2)

Den vollständigen Code zum Kopieren in die Google Ads Skript-Umgebung findest du hier. Der Einstieg ist die Funktion main() am Anfang. Alle relevanten Einstellungen sind in der CONFIG-Sektion gebündelt.

```
/**
 * =====
 * Google Ads MCC-Skript: Bidding-Anomalie-Detektor
 * =====
 *
 * ZWECK:
 * Erkennt Kampagnen, die vom Google-Bidding-Update am 17. August 2026 betroffen
 * sein könnten. Konkret: budget-limitierte Kampagnen mit Target CPA oder Target
 * ROAS, die ihren gesetzten Zielwert aktuell signifikant übertreffen – diese
 * werden nach dem Update näher am eingestellten Ziel performen, was einen
 * Performance-Einbruch bedeuten kann, wenn die Targets nicht angepasst werden.
 *
 * SETUP (3 Schritte):
 * 1. Google Sheet anlegen mit folgender Struktur (Zeile 1 = Header, ab Zeile 2 Daten):
 *   Spalte A: Account-ID      (Format: XXX-XXX-XXXX oder nur Zahlen)
 *   Spalte B: Account-Name    (beliebiger Label, nur zur Übersicht)
 *   Spalte C: E-Mail(s)       (kommagetrennt, z.B. "max@agentur.de,kunde@brand.de")
 *   Spalte D: Aktiv           (TRUE = wird verarbeitet, FALSE = wird übersprungen)
 *
 * 2. Sheet-URL in CONFIG.SHEET_URL eintragen (s.u.)
 *
 * 3. Skript im MCC-Konto unter Tools → Skripte hinzufügen und planen
 *   (z.B. wöchentlich montags, damit man rechtzeitig vor dem 17.08. handeln kann)
 *
 * BETROFFENE KAMPAGNENTYPEN laut Google-Ankündigung:
 *  Search, Shopping, Performance Max, Demand Gen, Display, Travel
 *  App-Kampagnen, Video-Reichweite, VVC (nicht betroffen, werden übersprungen)
 *
 * -----
 * ÄNDERUNGEN ggü. v1 (siehe Kommentare "// [v2]"):
 * 2. Analysezeitraum: 28 statt 30 Tage (Alignment mit Google-Smart-Bidding-Fenster)
 * 3. Zwei Report-Modi: 'TODOS_ONLY' (Mail nur bei Handlungsbedarf) & 'ALL' (immer)
 * 4. Ist-Werte-Bug behoben: Kosten/Conv./Conv.-Value kommen jetzt zuverlässig
 *   per GAQL (stats.getConversionValue()) existiert NICHT im Scripts-SDK und
 *   lieferte darum immer 0 → Ist-ROAS 0,00).
 * 5. Fokus auf Handlungsbedarf: On-Target/Info-Kampagnen sind aus der Haupt-
 *   übersicht raus und stehen kompakt am Ende (bzw. in TODOS_ONLY ganz weg).
 * 6. Kampagnen-Deep-Link repariert (korrektes __e-Format, kein untaugliches ocid).
 * 7. Scope-Diskrepanz: Budget-Limitierung wird jetzt über das ganze Fenster +
 *   "Budget verlorener IS" erkannt (nicht nur der Vortag). PMax/alle Typen laufen
 *   einheitlich über GAQL. Ausgeschlossene Kampagnen werden protokolliert
 *   (Diagnose-Block), damit Abweichungen zum Google-Tool nachvollziehbar sind.
 * =====
 */

// =====
// CONFIG – hier anpassen
// =====
var CONFIG = {
  // URL des Google Sheets mit Account-Konfiguration
  SHEET_URL:
'https://docs.google.com/spreadsheets/d/1skydq34u00Zn90njcgTBF3D7HlnWUMDNx7HQfXaUa00/edit?gid=0#gid=0',
  SHEET_NAME: 'Config', // Tab-Name im Sheet

  // [v2 - Punkt 2] Analysezeitraum (Tage zurück).
  // 28 Tage = 4 volle Wochen und deckt sich mit dem Fenster, das Googles
  // Smart Bidding zur Optimierung heranzieht (glättet Wochentag-Effekte).
  LOOKBACK_DAYS: 28,

  // [v2 - Punkt 3] Report-Modus:
```

```

// 'TODOS_ONLY' → E-Mail wird NUR verschickt, wenn es echten Handlungsbedarf
// gibt (mind. 1 Kampagne mit Ziel deutlich übertroffen).
// Info-/On-Target-Kampagnen werden nicht gezeigt.
// 'ALL' → E-Mail wird IMMER verschickt (auch "alles auf Ziel"),
// inkl. kompakter Info- und Diagnose-Abschnitte am Ende.
// Tipp: Zwei Zeitpläne mit je einer Kopie des Skripts anlegen, wenn beide
// Modi parallel gebraucht werden (z.B. TODOS_ONLY wöchentlich an den
// Kunden, ALL monatlich intern).
REPORT_MODE: 'TODOS_ONLY',

// Schwellenwert: CPA-Überperformance
// Alert (Handlungsbedarf) wenn: Ist-CPA < Ziel-CPA * (1 - CPA_GAP_THRESHOLD)
// 0.10 = Alert, wenn der Ist-CPA mehr als 10 % UNTER dem Ziel liegt.
CPA_GAP_THRESHOLD: 0.10,

// Schwellenwert: ROAS-Überperformance
// Alert (Handlungsbedarf) wenn: Ist-ROAS > Ziel-ROAS * (1 + ROAS_GAP_THRESHOLD)
// 0.10 = Alert, wenn der Ist-ROAS mehr als 10 % ÜBER dem Ziel liegt.
ROAS_GAP_THRESHOLD: 0.10,

// Ab diesem Gap gilt der Fall als HIGH (sonst MEDIUM).
HIGH_SEVERITY_GAP: 0.50,

// Mindest-Conversions für statistisch verlässliche CPA/ROAS-Berechnung
MIN_CONVERSIONS: 5,

// [v2 - Punkt 7] Budget-Limitierung.
// Signal A: "Budget verlorener Impr.-Anteil" (Search/Shopping/Display) ≥ diesem Wert.
BUDGET_LOST_IS_THRESHOLD: 0.05, // 5 % Impressionen wegen Budget verloren
// Signal B (Fallback, gilt für alle Typen inkl. PMax): durchschnittliche
// Tagesausgabe über den Zeitraum ≥ dieser Auslastung des Tagesbudgets.
BUDGET_LIMIT_THRESHOLD: 0.80, // 80 % des Tagesbudgets im Schnitt

// In TODOS_ONLY zusätzlich zu den Handlungsbedarf-Karten auch "Zur Prüfung"
// (zu wenig Daten) mit anzeigen, WENN ohnehin eine Mail rausgeht.
TODOS_INCLUDE_CHECK: true,

// Betreff-Präfix für Alert-Mails
EMAIL_SUBJECT_PREFIX: 'Google Ads Alert Bidding-Änderung 17.08.2026',

// Währungssymbol für die E-Mail-Formatierung
CURRENCY_SYMBOL: '€',
};

// MCC-Account-ID (gesetzt in main() bevor child accounts selektiert werden)
// Wird für Google Ads Deep-Links in E-Mails verwendet.
var MCC_CUSTOMER_ID = '574-260-3578';

// =====
// HAUPTFUNKTION
// =====
function main() {
  // MCC-ID einmalig vor dem Wechsel in child accounts speichern.
  MCC_CUSTOMER_ID = AdsApp.currentAccount().getCustomerId().replace(/-/g, '');
  Logger.log('MCC-ID: ' + MCC_CUSTOMER_ID);
  Logger.log('=== Bidding-Anomalie-Detektor (v2) gestartet ===');
  Logger.log('Modus: ' + CONFIG.REPORT_MODE + ' | Zeitraum: ' + CONFIG.LOOKBACK_DAYS + '
Tage');

  var sheet = getConfigSheet();
  if (!sheet) return;

  var accountConfigs = readAccountConfigs(sheet);
  if (accountConfigs.length === 0) {
    Logger.log('Keine aktiven Accounts im Config-Sheet gefunden.');
```

```

    }
  }
  Logger.log('\n=== Fertig ===');
}

// =====
// ACCOUNT VERARBEITEN
// =====
function processAccount(config) {
  var accountIterator = AdsManagerApp.accounts()
    .withIds([config.accountId])
    .get();
  if (!accountIterator.hasNext()) {
    Logger.log('Account ' + config.accountId + ' nicht gefunden oder kein Zugriff.');
```

return;

```
  }
  var account = accountIterator.next();
  AdsManagerApp.select(account);

  var accountCustomerId = AdsApp.currentAccount().getCustomerId(); // z.B. 123-456-7890
  var dateRange = getDateRange(CONFIG.LOOKBACK_DAYS);

  // [v2 - Punkt 4/7] Alle Daten via GAQL laden (zuverlässige Ist-Werte, alle
  // Kampagnentypen inkl. PMax in EINEM Durchlauf):
  //   biddingMap → Bidding-Strategie + Ziel + Budget + Kanaltyp je Kampagne
  //   metricsMap → Kosten, Conv., Conv.-Value, Budget-verlorener-IS über den Zeitraum
  // [Timing] Laufzeit je Phase protokollieren, um Flaschenhälse zu finden.
  var t0 = nowMs();
  var biddingMap = buildBiddingInfoMap();
  var t1 = nowMs();
  var metricsMap = buildMetricsMap(dateRange);
  var t2 = nowMs();
  Logger.log(' Bidding-Map: ' + Object.keys(biddingMap).length + ' Kampagnen (' +
secs(t0, t1) + ') | ' +
    'Metrics-Map: ' + Object.keys(metricsMap).length + ' Kampagnen (' + secs(t1,
t2) + ')');

  var anomalies = []; // Kampagnen mit relevanter Aussage (ACTION/CHECK/INFO)
  var diagnostics = []; // ziel-basierte Kampagnen, die BEWUSST rausgefallen sind

  for (var cid in biddingMap) {
    var bidding = biddingMap[cid];

    // App-/Video-Kampagnen überspringen (nicht betroffen laut Google)
    if (isExcludedChannelType(bidding.channelType)) continue;

    // Ziel-basiert? Wenn Strategie tCPA/tROAS ist, das Ziel aber nicht ermittelt
    // werden konnte (z.B. Portfolio ohne Zugriff), als Diagnose festhalten statt
    // stillschweigend zu ignorieren (Punkt 7 - Abweichung zum Google-Tool).
    if (!bidding.isTargetBased) {
      if (bidding.type === 'TARGET_CPA' || bidding.type === 'TARGET_ROAS') {
        diagnostics.push(makeDiag(bidding, metricsMap[cid], 'ZIEL_NICHT_ERMITTELT'));
      }
      continue;
    }

    var result = analyzeCampaignData(cid, bidding, metricsMap[cid], accountCustomerId);
    if (result.anomaly) {
      anomalies.push(result.anomaly);
    } else if (result.diag) {
      diagnostics.push(result.diag);
    }
  }

  Logger.log(' Analyse-Loop: ' + secs(t2, nowMs()) + ' über ' +
    Object.keys(biddingMap).length + ' Kampagnen.');
```

// Kategorisieren

```
var actions = anomalies.filter(function(a) { return a.category === 'ACTION'; });
var checks = anomalies.filter(function(a) { return a.category === 'CHECK'; });
var infos = anomalies.filter(function(a) { return a.category === 'INFO'; });

// Nach Dringlichkeit sortieren (größter Gap zuerst)
actions.sort(function(a, b) { return maxGapOf(b) - maxGapOf(a); });

Logger.log(' Ergebnis: ' + actions.length + ' Handlungsbedarf | ' +
  checks.length + ' Zur Prüfung | ' + infos.length + ' Auf Ziel | ' +
```

```

        diagnostics.length + ' ausgeschlossen (Diagnose).');

// [v2 - Punkt 3/5] Versand-Entscheidung je Modus
var mode = CONFIG.REPORT_MODE;
var shown;
if (mode === 'TODOS_ONLY') {
    if (actions.length === 0) {
        Logger.log(' TODOS_ONLY: kein Handlungsbedarf – keine Mail versendet.');
```

return;

```
    }
    shown = actions.concat(CONFIG.TODOS_INCLUDE_CHECK ? checks : []);
} else { // 'ALL'
    shown = actions.concat(checks).concat(infos);
}

sendAlertEmail(config, account.getName(), accountCustomerId, dateRange, {
    mode: mode,
    actions: actions,
    checks: checks,
    infos: infos,
    shown: shown,
    diagnostics: diagnostics,
});
}

// =====
// EINZELNE KAMPAGNE ANALYSIEREN (rein datenbasiert, keine Iteratoren)
// Rückgabe: { anomaly } ODER { diag } ODER {}
// =====
function analyzeCampaignData(campaignId, bidding, metrics, accountCustomerId) {
    metrics = metrics || { cost: 0, conversions: 0, conversionValue: 0, clicks: 0,
    impressions: 0, budgetLostIS: null };

    var budgetLimited = isBudgetLimited(metrics, bidding.budgetAmount);

    // Stufe 1: Nicht budget-limitiert → laut Google nicht betroffen.
    if (!budgetLimited) {
        return { diag: makeDiag(bidding, metrics, 'NICHT_BUDGET_LIMITIERT') };
    }
    // Kein Spend → nichts zu berechnen.
    if (metrics.cost <= 0) {
        return { diag: makeDiag(bidding, metrics, 'KEIN_SPEND') };
    }

    var actualValue = 0;
    var targetValue = bidding.targetValue;
    var gap = 0;
    var metricLabel = '';
    var targetLabel = '';
    var actualLabel = '';
    var threshold = CONFIG.CPA_GAP_THRESHOLD;

    if (bidding.type === 'TARGET_CPA' && targetValue > 0 && metrics.conversions > 0) {
        actualValue = metrics.cost / metrics.conversions;
        gap = (targetValue - actualValue) / targetValue; // positiv = Ist besser (günstiger)
als Ziel
        metricLabel = 'CPA';
        threshold = CONFIG.CPA_GAP_THRESHOLD;
        targetLabel = formatCurrency(targetValue);
        actualLabel = formatCurrency(actualValue);
    } else if (bidding.type === 'TARGET_ROAS' && targetValue > 0) {
        actualValue = metrics.conversionValue / metrics.cost; // jetzt korrekt (GAQL-Wert)
        gap = (actualValue - targetValue) / targetValue; // positiv = Ist besser (höher) als
Ziel
        metricLabel = 'ROAS';
        threshold = CONFIG.ROAS_GAP_THRESHOLD; // [v2] eigener ROAS-Schwellenwert (Bug in v1)
        targetLabel = formatRoas(targetValue);
        actualLabel = formatRoas(actualValue);
    } else {
        // Ziel-Typ bekannt, aber (noch) keine belastbare Basis
        return { diag: makeDiag(bidding, metrics, 'KEINE_CONVERSIONS') };
    }

    var hasEnoughData = metrics.conversions >= CONFIG.MIN_CONVERSIONS;

    var issues = [];
    var category;
```

```

if (!hasEnoughData) {
  // Zur Prüfung: budget-limitiert, aber Datenbasis zu dünn
  category = 'CHECK';
  issues.push({
    type: 'UNZUREICHENDE_DATEN',
    severity: 'LOW',
    detail: 'Budget-limitiert, aber nur ' + metrics.conversions.toFixed(1) +
      ' Conversions in ' + CONFIG.LOOKBACK_DAYS +
      ' Tagen - zu wenig für verlässliche ' + metricLabel + '-Analyse.',
    recommendation: 'Manuell im Bid Target Adjustment Tool prüfen (ab 6. Juli 2026).',
    dualPath: false,
    actualValue: actualValue, targetValue: targetValue, gap: gap,
  });
} else if (gap >= threshold) {
  // Handlungsbedarf: Ist schlägt Ziel deutlich → Kernfall des Flowcharts
  category = 'ACTION';
  var severity = gap >= CONFIG.HIGH_SEVERITY_GAP ? 'HIGH' : 'MEDIUM';
  issues.push({
    type: metricLabel + '_ZIEL_ÜBERTROFFEN',
    severity: severity,
    detail: 'Ziel-' + metricLabel + ': ' + targetLabel +
      ' | Ist-' + metricLabel + ': ' + actualLabel +
      ' | Überperformance: ' + formatPercent(gap),
    dualPath: true,
    pathA: {
      label: '🔴 Ziel ist echtes Geschäftsziel - Effizienz halten',
      action: 'Ziel auf Ist-Wert (' + actualLabel + ') anpassen via Bid Target Adjustment Tool ' +
        '<strong>vor dem 17.08.2026</strong>.',
      alt: 'Alternativ nichts tun: Performance driftet nach dem 17.08. zum gesetzten Ziel (' +
        targetLabel + ') - Volumen kann steigen, Effizienz sinkt.',
    },
    pathB: {
      label: '⚡ Ziel war als Puffer künstlich eng gesetzt',
      action: 'Auf den wahren Margin-Ceiling anheben (z.B. ' + targetLabel +
        ' war Puffer, echter Spielraum höher). Puffer-Trick funktioniert ab 17.08. nicht mehr.',
    },
    actualValue: actualValue, targetValue: targetValue, gap: gap,
  });
} else {
  // Info: bereits auf/nahe Ziel → minimaler Impact
  category = 'INFO';
  issues.push({
    type: 'BEREITS_AUF_ZIEL',
    severity: 'INFO',
    detail: 'Budget-limitiert, aber Ist-' + metricLabel + ' (' + actualLabel +
      ') liegt bereits nahe am gesetzten Ziel (' + targetLabel +
      '). Minimaler Impact nach 17.08. erwartet.',
    recommendation: 'Kein dringender Handlungsbedarf. Kampagne nach dem 17.08. beobachten.',
    dualPath: false,
    actualValue: actualValue, targetValue: targetValue, gap: gap,
  });
}

return {
  anomaly: {
    category: category,
    campaignName: bidding.name,
    campaignId: campaignId,
    accountId: accountCustomerId,
    campaignType: displayChannel(bidding.channelType),
    isBudgetLimited: true,
    bidding: bidding,
    issues: issues,
    stats: {
      cost: metrics.cost,
      conversions: metrics.conversions,
      conversionValue: metrics.conversionValue,
      clicks: metrics.clicks,
      impressions: metrics.impressions,
    },
  },
};

```

```

}

// Diagnose-Objekt für ausgeschlossene Kampagnen (Punkt 7)
function makeDiag(bidding, metrics, reason) {
  metrics = metrics || { cost: 0, conversions: 0 };
  return {
    name: bidding.name,
    type: bidding.displayType || bidding.type,
    target: bidding.targetValue > 0
      ? (bidding.type === 'TARGET_CPA' ? formatCurrency(bidding.targetValue) :
formatRoas(bidding.targetValue))
      : '-',
    cost: metrics.cost || 0,
    conversions: metrics.conversions || 0,
    reason: reason,
    reasonLabel: DIAG_LABELS[reason] || reason,
  };
}

var DIAG_LABELS = {
  NICHT_BUDGET_LIMITIERT: 'Nicht budget-limitiert (laut Google nicht betroffen)',
  KEIN_SPEND: 'Kein Spend im Zeitraum',
  KEINE_CONVERSIONS: 'Budget-limitiert, aber 0 Conversions/Conv.-Value',
  ZIEL_NICHT_ERMITTELT: 'tCPA/tROAS erkannt, Zielwert nicht auslesbar
(Portfolio/Zugriff?)',
};

// =====
// BIDDING-MAP VIA GAQL AUFBAUEN (einmalig pro Account)
// =====
// WICHTIG - camelCase-Zugriff:
//   Google Ads Scripts gibt GAQL-Felder in camelCase zurück, auch wenn
//   sie im SELECT snake_case heißen.
// =====
function buildBiddingInfoMap() {
  var biddingMap = {};

  // Schritt 1: Alle aktiven Kampagnen mit Bidding-, Kanal- und Budget-Feldern.
  var query =
    'SELECT ' +
    '  campaign.id, ' +
    '  campaign.name, ' +
    '  campaign.advertising_channel_type, ' +           // [v2] Kanaltyp
    (Video/App-Ausschluss)
    '  campaign.bidding_strategy_type, ' +
    '  campaign.target_cpa.target_cpa_micros, ' +
    '  campaign.target_roas.target_roas, ' +
    '  campaign.maximize_conversions.target_cpa_micros, ' +
    '  campaign.maximize_conversion_value.target_roas, ' +
    '  campaign_budget.amount_micros ' +             // [v2] Tagesbudget für
Limit-Check
    'FROM campaign ' +
    'WHERE campaign.status = 'ENABLED'';

  try {
    var rows = AdsApp.search(query);
    while (rows.hasNext()) {
      var row = rows.next();
      var id = String(row.campaign.id);
      var type = row.campaign.biddingStrategyType || '';
      var channelType = row.campaign.advertisingChannelType || '';
      var budgetAmount = (row.campaignBudget && row.campaignBudget.amountMicros)
        ? Number(row.campaignBudget.amountMicros) / 1000000 : 0;

      var info = {
        name: row.campaign.name,
        channelType: channelType,
        isTargetBased: false,
        type: type,
        displayType: type,
        targetValue: 0,
        budgetAmount: budgetAmount,
      };

      if (type === 'TARGET_CPA') {
        var micros = (row.campaign.targetCpa && row.campaign.targetCpa.targetCpaMicros)
|| 0;

```

```

        if (micros > 0) { info.isTargetBased = true; info.targetValue = Number(micros) /
1000000; }
        } else if (type === 'TARGET_ROAS') {
            var roas = (row.campaign.targetRoas && row.campaign.targetRoas.targetRoas) || 0;
            if (roas > 0) { info.isTargetBased = true; info.targetValue = Number(roas); }
        } else if (type === 'MAXIMIZE_CONVERSIONS') {
            var mcMicros = (row.campaign.maximizeConversions &&
row.campaign.maximizeConversions.targetCpaMicros) || 0;
            if (mcMicros > 0) {
                info.isTargetBased = true; info.type = 'TARGET_CPA';
                info.displayType = 'TARGET_CPA (Max. Conversions)';
                info.targetValue = Number(mcMicros) / 1000000;
            }
        } else if (type === 'MAXIMIZE_CONVERSION_VALUE') {
            var mcvRoas = (row.campaign.maximizeConversionValue &&
row.campaign.maximizeConversionValue.targetRoas) || 0;
            if (mcvRoas > 0) {
                info.isTargetBased = true; info.type = 'TARGET_ROAS';
                info.displayType = 'TARGET_ROAS (Max. Conv. Value)';
                info.targetValue = Number(mcvRoas);
            }
        }
    }

    biddingMap[id] = info;
}
} catch (e) {
    Logger.log('GAQL-Fehler Schritt 1 (Bidding): ' + e);
}

// Schritt 2: Portfolio-Strategien nachladen (Ziel liegt auf geteilter Strategie).
try {
    var portfolioIds = [];
    for (var cid in biddingMap) {
        var entry = biddingMap[cid];
        if ((entry.type === 'TARGET_CPA' || entry.type === 'TARGET_ROAS') &&
entry.targetValue === 0) {
            portfolioIds.push(cid);
        }
    }
    if (portfolioIds.length > 0) {
        var tp = nowMs();
        Logger.log(' Portfolio-Nachladen für ' + portfolioIds.length + ' Kampagne(n)...');
        var pQuery =
            'SELECT ' +
            ' campaign.id, campaign.name, ' +
            ' bidding_strategy.type, ' +
            ' bidding_strategy.target_cpa.target_cpa_micros, ' +
            ' bidding_strategy.target_roas.target_roas ' +
            'FROM campaign ' +
            'WHERE campaign.status = 'ENABLED' AND campaign.id IN (" + portfolioIds.join(',')
+ ')';
        var pRows = AdsApp.search(pQuery);
        while (pRows.hasNext()) {
            var pRow = pRows.next();
            var pid = String(pRow.campaign.id);
            var bsType = (pRow.biddingStrategy && pRow.biddingStrategy.type) || '';
            var bsMicros = (pRow.biddingStrategy && pRow.biddingStrategy.targetCpa &&
pRow.biddingStrategy.targetCpa.targetCpaMicros) || 0;
            var bsRoas = (pRow.biddingStrategy && pRow.biddingStrategy.targetRoas &&
pRow.biddingStrategy.targetRoas.targetRoas) || 0;
            if (biddingMap[pid]) {
                if (bsType === 'TARGET_CPA' && bsMicros > 0) {
                    biddingMap[pid].isTargetBased = true;
                    biddingMap[pid].targetValue = Number(bsMicros) / 1000000;
                    biddingMap[pid].displayType = 'TARGET_CPA (Portfolio)';
                } else if (bsType === 'TARGET_ROAS' && bsRoas > 0) {
                    biddingMap[pid].isTargetBased = true;
                    biddingMap[pid].targetValue = Number(bsRoas);
                    biddingMap[pid].displayType = 'TARGET_ROAS (Portfolio)';
                }
            }
        }
        Logger.log(' Portfolio-Nachladen fertig (' + secs(tp, nowMs()) + ').');
    }
} catch (e) {
    Logger.log('Portfolio-Nachladen fehlgeschlagen: ' + e);
}

```

```

    return biddingMap;
}

// =====
// [v2 - Punkt 4/7] METRICS-MAP VIA GAQL (zuverlässige Ist-Werte)
// Ersetzt campaign.getStatsFor() + stats.getConversionValue()
// (getConversionValue() existiert NICHT im Scripts-SDK → lieferte immer 0).
// =====
function buildMetricsMap(dateRange) {
    var map = {};

    var baseSelect =
        ' campaign.id, ' +
        ' metrics.cost_micros, ' +
        ' metrics.conversions, ' +
        ' metrics.conversions_value, ' +
        ' metrics.clicks, ' +
        ' metrics.impressions';
    var where =
        "FROM campaign WHERE campaign.status = 'ENABLED' " +
        "AND segments.date BETWEEN '" + dateRange.startGaql + "' AND '" + dateRange.endGaql +
        """;

    // Erst mit Budget-verlorenem-IS versuchen; falls das Feld die Query sprengt,
    // ohne es erneut versuchen (Fallback rein auf Budget-/Spend-Ratio).
    var withIS = 'SELECT ' + baseSelect + ', metrics.search_budget_lost_impression_share '
+ where;
    var withoutIS = 'SELECT ' + baseSelect + ' ' + where;

    var rows = null, hasIS = true;
    try {
        rows = AdsApp.search(withIS);
    } catch (e) {
        Logger.log(' Hinweis: search_budget_lost_impression_share nicht abfragbar (' + e +
        ') - Fallback ohne IS.');
```

```

        hasIS = false;
        try { rows = AdsApp.search(withoutIS); } catch (e2) { Logger.log('GAQL-Fehler
Metrics: ' + e2); return map; }
    }

    while (rows.hasNext()) {
        var row = rows.next();
        var id = String(row.campaign.id);
        var m = row.metrics || {};
        map[id] = {
            cost: (m.costMicros ? Number(m.costMicros) / 1000000 : 0),
            conversions: (m.conversions ? Number(m.conversions) : 0),
            conversionValue: (m.conversionsValue ? Number(m.conversionsValue) : 0),
            clicks: (m.clicks ? Number(m.clicks) : 0),
            impressions: (m.impressions ? Number(m.impressions) : 0),
            budgetLostIS: (hasIS && m.searchBudgetLostImpressionShare != null)
                ? Number(m.searchBudgetLostImpressionShare) : null,
        };
    }
    return map;
}

// =====
// [v2 - Punkt 7] BUDGET-LIMITIERUNG PRÜFEN (Fenster statt nur Vortag)
// =====
function isBudgetLimited(metrics, budgetAmount) {
    if (!metrics) return false;
    // Signal A: Google meldet Impressionsverlust wegen Budget (Search/Shopping/Display)
    if (metrics.budgetLostIS != null && metrics.budgetLostIS >=
CONFIG.BUDGET_LOST_IS_THRESHOLD) {
        return true;
    }
    // Signal B (Fallback für alle Typen inkl. PMax): Ø-Tagesausgabe vs. Tagesbudget
    if (budgetAmount > 0) {
        var avgDaily = metrics.cost / CONFIG.LOOKBACK_DAYS;
        if ((avgDaily / budgetAmount) >= CONFIG.BUDGET_LIMIT_THRESHOLD) return true;
    }
    return false;
}

// =====

```



```
// KAMPAGNENTYPEN AUSSCHLIESSEN (nicht betroffen laut Google)
// =====
function isExcludedChannelType(channelType) {
    var excluded = ['VIDEO', 'MULTI_CHANNEL']; // MULTI_CHANNEL = App-Kampagnen
    if (!channelType) return false;
    var c = String(channelType).toUpperCase();
    for (var i = 0; i < excluded.length; i++) {
        if (c.indexOf(excluded[i]) !== -1) return true;
    }
    return false;
}

// Lesbarer Kanaltyp für die Anzeige
function displayChannel(channelType) {
    var map = {
        SEARCH: 'Search', SHOPPING: 'Shopping', DISPLAY: 'Display',
        PERFORMANCE_MAX: 'Performance Max', DEMAND_GEN: 'Demand Gen',
        DISCOVERY: 'Demand Gen', TRAVEL: 'Travel', HOTEL: 'Hotel',
        LOCAL: 'Local', SMART: 'Smart',
    };
    return map[String(channelType).toUpperCase()] || (channelType || 'Kampagne');
}

// =====
// [v2 - Punkt 6] GOOGLE ADS DEEP-LINK
// =====
// Repariert. Bisheriger Bug: als "ocid" wurde die numerische Kundennummer
// gesetzt - "ocid" ist aber eine OPAKE ID und NICHT die 10-stellige Kundennr.,
// darum landete der Link auf der allgemeinen Kampagnenübersicht.
// Korrekt: "__e" = das aufzurufende (Client-)Konto, 10-stellig OHNE Bindestriche.
// Voraussetzung: Nutzer ist in Chrome mit einem Konto eingeloggt, das Zugriff
// auf dieses Kundenkonto hat (i.d.R. der MCC-Login).
// =====
function buildCampaignUrl(accountCustomerId, campaignId) {
    var clientId = String(accountCustomerId || '').replace(/-/g, '');
    return 'https://ads.google.com/aw/campaigns' +
        '?campaignId=' + campaignId +
        (clientId ? '&__e=' + clientId : '');
}

// =====
// norisk CI - Brand-Farben & Typografie
// =====
var NR = {
    dark: '#242222', coral: '#D65A47', yellow: '#F0F37E', blue: '#4A72C7',
    beige: '#E1D4B5', light: 'EEEEEE', white: 'FFFFFF', text: '#242222',
    muted: '#888888', font: '"Inter","Helvetica Neue",Arial,sans-serif',
};

// =====
// E-MAIL ALERT SENDEN
// =====
function sendAlertEmail(config, accountName, accountCustomerId, dateRange, data) {
    var subject = buildSubject(accountName, data);
    var body = buildEmailBody(accountName, config.accountId, accountCustomerId, dateRange, data);

    for (var i = 0; i < config.emails.length; i++) {
        MailApp.sendEmail({ to: config.emails[i], subject: subject, htmlBody: body });
        Logger.log(' Alert gesendet an: ' + config.emails[i]);
    }
}

function buildSubject(accountName, data) {
    var p = CONFIG.EMAIL_SUBJECT_PREFIX + ' - ' + accountName + ': ';
    if (data.actions.length > 0) {
        return p + data.actions.length + ' Kampagne(n) mit Handlungsbedarf';
    }
    return p + 'kein Handlungsbedarf (alles auf Ziel)';
}

function getMaxSeverity(anomaly) {
    var order = { HIGH: 4, MEDIUM: 3, LOW: 2, INFO: 1 };
    var max = 'INFO';
    for (var i = 0; i < anomaly.issues.length; i++) {
        var s = anomaly.issues[i].severity;
        if ((order[s] || 0) > (order[max] || 0)) max = s;
    }
}
```

```

    }
    return max;
}

function maxGapOf(anomaly) {
    var g = 0;
    for (var i = 0; i < anomaly.issues.length; i++) {
        if ((anomaly.issues[i].gap || 0) > g) g = anomaly.issues[i].gap;
    }
    return g;
}

// =====
// E-MAIL-BODY - Gmail-kompatibles Table-Layout, alle Styles inline
// =====
function buildEmailBody(accountName, accountId, accountCustomerId, dateRange, data) {
    var html = '<!DOCTYPE html><html><head><meta charset="UTF-8">' +
        '<meta name="viewport" content="width=device-width,initial-scale=1"></head>'
    +
        '<body style="margin:0;padding:0;background-color:#F5F4F0;" ' +
        'font-family:Arial,Helvetica,sans-serif;">' +
        '<table width="100%" cellpadding="0" cellspacing="0" border="0" ' +
        'style="background-color:#F5F4F0;">' +
        '<tr><td align="center" style="padding:24px 16px;">' +
        '<table width="600" cellpadding="0" cellspacing="0" border="0" ' +
        'style="max-width:600px;width:100%;">';

    // --- Header ---
    html += '<tr><td bgcolor="' + NR.dark + '" style="background-color:' + NR.dark + ';" ' +
        'padding:24px 28px 20px;border-radius:6px 6px 0 0;">' +
        '<p style="margin:0 0 8px 0;font-size:11px;letter-spacing:2px;color:#888888;' +
        'text-transform:uppercase;font-family:Arial,Helvetica,sans-serif;">' +
        'norisk group &middot; Performance Marketing</p>' +
        '<p style="margin:0;font-size:22px;font-weight:700;color:#ffffff;' +
        'font-family:Arial,Helvetica,sans-serif;line-height:1.2;">Google Ads
    Bidding-Report</p>' +
        '<p style="margin:4px 0 12px 0;font-size:20px;font-weight:400;color:' +
    NR.coral + ';" ' +
        'font-family:Arial,Helvetica,sans-serif;">' + escapeHtml(accountName) + '</p>'
    +
        '<p
    style="margin:0;font-size:12px;color:#AAAAAA;font-family:Arial,Helvetica,sans-serif;">' +
        'Analyse: ' + formatDateDisplay(dateRange.startDate) + ' &ndash; ' +
        formatDateDisplay(dateRange.endDate) +
        ' (' + CONFIG.LOOKBACK_DAYS + ' Tage) &nbsp;&nbsp;&middot;&nbsp;&nbsp;&nbsp; Account-ID: ' +
        accountId + '</p>' +
        '</td></tr>';

    // --- Warnbanner ---
    html += '<tr><td bgcolor="' + NR.yellow + '" style="background-color:' + NR.yellow +
    ';" ' +
        'padding:14px 28px;border-left:5px solid ' + NR.dark + ';">' +
        '<table width="100%" cellpadding="0" cellspacing="0" border="0"><tr>' +
        '<td style="font-family:Arial,Helvetica,sans-serif;">' +
        '<span style="font-size:13px;font-weight:700;color:' + NR.dark + ';">' +
        '&#9888; Handlungsbedarf bis 17. August 2026</span>' +
        '<span style="font-size:12px;color:' + NR.dark + ';"> &ndash; ' +
        'Budget-limitierte Kampagnen mit tCPA/tROAS werden ab dann strikt auf den
    Zielwert optimiert.' +
        '</span></td>' +
        '<td width="80" style="text-align:right;vertical-align:middle;">' +
        '<a href="https://support.google.com/google-ads/answer/17061251" ' +
        'style="color:' + NR.dark +
        ';"font-size:11px;font-weight:700;white-space:nowrap;">&#8594; Google</a>' +
        '</td></tr></table></td></tr>';

    // --- Summary-Kacheln ---
    html += '<tr><td style="background-color:#ffffff;border-bottom:1px solid ' + NR.light +
    ';">' +
        '<table width="100%" cellpadding="0" cellspacing="0" border="0"><tr>';
    html += summaryCard('Handlungsbedarf', data.actions.length, '#ffffff', NR.coral);
    html += summaryCard('Zur Pr&uuml;fung', data.checks.length, NR.dark, NR.yellow);
    html += summaryCard('Auf Ziel', data.infos.length, '#ffffff', NR.blue);
    html += summaryCard('Ausgeschl.', data.diagnostics.length, NR.dark, NR.beige);
    html += '</tr></table></td></tr>';

    // --- "Alles auf Ziel"-Hinweis (nur ALL-Modus ohne Handlungsbedarf) ---

```

```

    if (data.actions.length === 0) {
        html += '<tr><td bgcolor="#ffffff" style="background-color:#ffffff;padding:18px
24px;' +
            'border-bottom:1px solid ' + NR.light + ';' +
            '<p style="margin:0;font-size:13px;color:#2A7A20;font-weight:700;' +
            'font-family:Arial,Helvetica,sans-serif;">&#10003; Kein akuter
Handlungsbedarf</p>' +
            '<p style="margin:4px 0 0 0;font-size:12px;color:' + NR.muted + ';' +
            'font-family:Arial,Helvetica,sans-serif;">Keine budget-limitierte
tCPA/tROAS-Kampagne ' +
            'übertrifft ihr Ziel über dem Schwellenwert.</p></td></tr>';
    }

    // — Performance-Chart (nur Fokus-Kampagnen: Handlungsbedarf + Prüfung) —
    var chartItems = data.actions.concat(data.checks);
    var chart = generatePerformanceChart(chartItems);
    if (chart) {
        html += '<tr><td bgcolor="#ffffff" style="background-color:#ffffff;padding:20px
24px;' +
            'border-bottom:1px solid ' + NR.light + ';' +
            '<p style="margin:0 0 14px 0;font-size:13px;font-weight:700;color:' + NR.dark
+ ';' +
            'font-family:Arial,Helvetica,sans-serif;">Ist-Performance vs. Zielwert
&mdash; Break-Even-Analyse</p>' +
            '<div style="overflow-x:auto;">' + chart + '</div></td></tr>';
    }

    // — Handlungsbedarf-Karten —
    if (data.actions.length > 0) {
        html += sectionHeader('Handlungsbedarf', NR.coral);
        html += '<tr><td bgcolor="#ffffff" style="background-color:#ffffff;padding:4px 24px
16px;">';
        for (var i = 0; i < data.actions.length; i++) html +=
buildCampaignCard(data.actions[i]);
        html += '</td></tr>';
    }

    // — Zur-Prüfung-Karten (nur wenn sichtbar laut Modus) —
    var showChecks = data.shown.filter(function(a) { return a.category === 'CHECK'; });
    if (showChecks.length > 0) {
        html += sectionHeader('Zur Prüfung – zu wenig Daten', NR.blue);
        html += '<tr><td bgcolor="#ffffff" style="background-color:#ffffff;padding:4px 24px
16px;">';
        for (var c = 0; c < showChecks.length; c++) html += buildCampaignCard(showChecks[c]);
        html += '</td></tr>';
    }

    // — Info / Auf Ziel – kompakt ans Ende (nur ALL-Modus) —
    if (data.mode === 'ALL' && data.infos.length > 0) {
        html += sectionHeader('Bereits auf Ziel – kein dringender Handlungsbedarf',
NR.muted);
        html += '<tr><td bgcolor="#ffffff" style="background-color:#ffffff;padding:4px 24px
12px;">' +
            '<table width="100%" cellpadding="0" cellspacing="0" border="0">';
        for (var k = 0; k < data.infos.length; k++) html += buildCompactRow(data.infos[k]);
        html += '</table></td></tr>';
    }

    // — Diagnose: ausgeschlossene ziel-basierte Kampagnen (Punkt 7) – nur ALL —
    if (data.mode === 'ALL' && data.diagnostics.length > 0) {
        html += sectionHeader('Diagnose – nicht im Scope & Grund', NR.muted);
        html += '<tr><td bgcolor="#ffffff" style="background-color:#ffffff;padding:4px 24px
16px;">' +
            '<p style="margin:0 0 8px 0;font-size:11px;color:' + NR.muted + ';' +
            'font-family:Arial,Helvetica,sans-serif;">Warum eine tCPA/tROAS-Kampagne
NICHT als ' +
            'Handlungsbedarf gelistet ist (hilft beim Abgleich mit dem
Google-Ads-Tool):</p>' +
            '<table width="100%" cellpadding="0" cellspacing="0" border="0">';
        for (var d = 0; d < data.diagnostics.length; d++) html +=
buildDiagRow(data.diagnostics[d]);
        html += '</table></td></tr>';
    }

    // — Footer —
    html += '<tr><td bgcolor="' + NR.dark + '" style="background-color:' + NR.dark + ';' +
            'padding:16px 28px;border-radius:0 0 6px 6px;">' +

```

```

        '<table width="100%" cellpadding="0" cellspacing="0" border="0"><tr>' +
        '<td style="font-family:Arial,Helvetica,sans-serif;">' +
        '<p style="margin:0;font-size:10px;color:#888888;">' +
        'Automatisch generiert &middot; Google Ads MCC Bidding-Anomalie-Detektor (v2)
&middot; Modus: ' +
        data.mode + '<br>Kampagnen-Anpassung via <strong style="color:#CCCCC;">Bid
Target Adjustment Tool</strong>' +
        ' in Google Ads (ab 6. Juli 2026)</p></td>' +
        '<td width="80" align="right" style="vertical-align:middle;">' +
        '<p style="margin:0;font-size:10px;color:' + NR.coral + ';font-weight:700;' +
        'font-family:Arial,Helvetica,sans-serif;">norisk.group</p></td>' +
        '</tr></table></td></tr>';

    html += '</table></td></tr></table></body></html>';
    return html;
}

// Abschnitts-Überschrift
function sectionHeader(title, color) {
    return '<tr><td bgcolor="#ffffff" style="background-color:#ffffff;padding:16px 24px
0;">' +
        '<p style="margin:0;font-size:13px;font-weight:700;color:' + NR.dark + ';' +
        'border-left:4px solid ' + color + ';padding-left:10px;' +
        'font-family:Arial,Helvetica,sans-serif;">' + escapeHtml(title) +
        '</p></td></tr>';
}

// summaryCard - <td> für Gmail-table-Layout
function summaryCard(label, count, textColor, bgColor) {
    return '<td width="25%" bgcolor="' + bgColor + '" ' +
        'style="background-color:' + bgColor + ';padding:16px 12px;text-align:center;' +
        'border-right:1px solid ' + NR.light + ';">' +
        '<p style="margin:0;font-size:26px;font-weight:700;color:' + textColor + ';' +
        'line-height:1;font-family:Arial,Helvetica,sans-serif;">' + count + '</p>' +
        '<p style="margin:5px 0 0 0;font-size:10px;color:' + textColor + ';' +
        'text-transform:uppercase;letter-spacing:1px;font-family:Arial,Helvetica,sans-serif;">' +
        label + '</p>' +
        '</td>';
}

// Kompakte Zeile für Info/On-Target-Kampagnen
function buildCompactRow(anomaly) {
    var issue = anomaly.issues[0] || {};
    var url = buildCampaignUrl(anomaly.accountId, anomaly.campaignId);
    return '<tr><td style="padding:7px 0;border-bottom:1px solid ' + NR.light + ';' +
        'font-family:Arial,Helvetica,sans-serif;">' +
        '<a href="' + url + '" target="_blank" style="font-size:12px;font-weight:700;' +
        'color:' + NR.dark + ';text-decoration:underline;">' +
        escapeHtml(anomaly.campaignName) + ' &#8599;</a>' +
        '<span style="font-size:11px;color:' + NR.muted + ';"> &nbsp;&middot;&nbsp;&nbsp;&nbsp;' +
        (issue.detail || '') + '</span></td></tr>';
}

// Diagnose-Zeile (ausgeschlossene Kampagne)
function buildDiagRow(diag) {
    return '<tr><td style="padding:6px 0;border-bottom:1px solid ' + NR.light + ';' +
        'font-family:Arial,Helvetica,sans-serif;">' +
        '<span style="font-size:12px;font-weight:700;color:' + NR.dark + ';">' +
        escapeHtml(diag.name) + '</span>' +
        '<span style="font-size:11px;color:' + NR.muted + ';"> &nbsp;&middot;&nbsp;&nbsp;&nbsp;' +
        diag.type + ' &middot; Ziel: ' + diag.target +
        ' &middot; Kosten: ' + formatCurrency(diag.cost) +
        ' &middot; Conv.: ' + diag.conversions.toFixed(1) + '</span><br>' +
        '<span style="font-size:11px;color:' + NR.coral + ';">&#8594;' +
        escapeHtml(diag.reasonLabel) + '</span>' +
        '</td></tr>';
}

// =====
// KAMPAGNEN-KARTE - Gmail-kompatibles Table-Layout
// =====
function buildCampaignCard(anomaly) {
    var maxSev = getMaxSeverity(anomaly);
    var sevColor = { HIGH: NR.coral, MEDIUM: '#B08800', LOW: NR.blue, INFO: NR.muted
}[maxSev] || NR.muted;

```

```

var sevBg = { HIGH: '#FAF0EE', MEDIUM: '#FAFCC9', LOW: '#EEF2FB', INFO: '#F5F5F5'
}[maxSev] || '#F5F5F5';
var sevLabel = { HIGH: 'HOHES RISIKO', MEDIUM: 'MITTLERES RISIKO', LOW: 'ZUR PRUEFUNG',
INFO: 'INFO' }[maxSev] || 'INFO';

var campaignUrl = buildCampaignUrl(anomaly.accountId, anomaly.campaignId);

var card = '<table width="100%" cellpadding="0" cellspacing="0" border="0" ' +
'style="border:1px solid ' + NR.light + ';border-left:4px solid ' + sevColor
+ ';' +
'margin-bottom:12px;border-collapse:collapse;">';

card += '<tr><td bgcolor="' + sevBg + '" style="background-color:' + sevBg +
';padding:11px 14px;' +
'border-bottom:1px solid ' + NR.light + ';">' +
'<table width="100%" cellpadding="0" cellspacing="0" border="0"><tr>' +
'<td style="font-family:Arial,Helvetica,sans-serif;">' +
'<a href="' + campaignUrl + '" target="_blank'
style="font-size:13px;font-weight:700;' +
'color:' + NR.dark + ';text-decoration:underline;">' +
escapeHtml(anomaly.campaignName) + ' &#8599;</a></td>' +
'<td width="130" align="right" valign="middle">' +
'<span style="font-size:9px;font-weight:700;color:' + sevColor +
';text-transform:uppercase;' +
'letter-spacing:0.5px;border:1px solid ' + sevColor + ';padding:2px 6px;' +
'font-family:Arial,Helvetica,sans-serif;">' + sevLabel +
'</span></td></tr></table>' +
'<p style="margin:5px 0 0 0;font-size:10px;color:' + NR.muted + ';' +
'font-family:Arial,Helvetica,sans-serif;">' +
anomaly.campaignType + ' &nbsp;&#8226;&nbsp;  Budget-limitiert: ' +
(anomaly.isBudgetLimited ? '<strong style="color:' + NR.coral +
';">JA</strong>' : 'Nein') +
' &nbsp;&#8226;&nbsp;  ' + (anomaly.bidding.displayType ||
anomaly.bidding.type) +
(anomaly.bidding.targetValue > 0
? ' &nbsp;&#8226;&nbsp;  Ziel: <strong>' + (anomaly.bidding.type ===
'TARGET_CPA'
? formatCurrency(anomaly.bidding.targetValue) :
formatRoas(anomaly.bidding.targetValue)) + '</strong>'
: '')) +
' &nbsp;&#8226;&nbsp;  Kosten (' + CONFIG.LOOKBACK_DAYS + 'T): ' +
formatCurrency(anomaly.stats.cost) +
' &nbsp;&#8226;&nbsp;  Conv.: ' + anomaly.stats.conversions.toFixed(1) +
'</p></td></tr>';

for (var i = 0; i < anomaly.issues.length; i++) {
var issue = anomaly.issues[i];
card += '<tr><td bgcolor="#ffffff" style="background-color:#ffffff;padding:12px
14px;">' +
'<p style="margin:0 0 6px
0;font-size:12px;font-family:Arial,Helvetica,sans-serif;">' +
'<strong style="color:' + NR.dark + ';">' + issue.type.replace(/_/g, ' ') +
'</strong>' +
(issue.detail ? '<span style="color:' + NR.muted + ';margin-left:8px;">' +
issue.detail + '</span>' : '') +
'</p>';

if (issue.dualPath) {
card += '<table width="100%" cellpadding="0" cellspacing="0" border="0"
style="margin-top:8px;"><tr>' +
'<td width="49%" valign="top" bgcolor="#EEF2FB"
style="background-color:#EEF2FB;' +
'border:1px solid ' + NR.blue + ';border-top:3px solid ' + NR.blue +
';padding:10px;' +
'font-size:11px;font-family:Arial,Helvetica,sans-serif;">' +
'<p style="margin:0 0 5px 0;font-weight:700;color:' + NR.blue + ';">' +
issue.pathA.label + '</p>' +
'<p style="margin:0 0 4px 0;color:' + NR.dark + ';"><strong>&#10003;
Effizienz halten:</strong><br>' +
issue.pathA.action + '</p>' +
'<p style="margin:0;color:' + NR.muted +
';font-style:italic;font-size:10px;">&#8594; Nichts tun: ' +
issue.pathA.alt + '</p></td>' +
'<td width="2%" style="font-size:1px;line-height:1px;">&nbsp;&#8203;</td>' +
'<td width="49%" valign="top" bgcolor="#FAF0EE"
style="background-color:#FAF0EE;' +

```

```

        'border:1px solid ' + NR.coral + ';border-top:3px solid ' + NR.coral +
';padding:10px;' +
        'font-size:11px;font-family:Arial,Helvetica,sans-serif;">' +
        '<p style="margin:0 0 5px 0;font-weight:700;color:' + NR.coral + ';">' +
issue.pathB.label + '</p>' +
        '<p style="margin:0;color:' + NR.dark + ';">' + issue.pathB.action +
'</p></td>' +
        '</tr></table>';
    } else if (issue.recommendation) {
        card += '<p style="margin:4px 0 0 0;font-size:11px;color:' + NR.blue + ';' +
        'font-family:Arial,Helvetica,sans-serif;">#8594; ' + issue.recommendation
+ '</p>';
    }
    card += '</td></tr>';
    if (i < anomaly.issues.length - 1) {
        card += '<tr><td style="height:1px;background-color:' + NR.light +
';font-size:0;line-height:0;">&nbsp;</td></tr>';
    }
    }
    card += '</table>';
    return card;
}

// =====
// PERFORMANCE CHART - Gmail-safe HTML-Table
// =====
function generatePerformanceChart(anomalies) {
    var chartData = [];
    for (var i = 0; i < anomalies.length; i++) {
        var a = anomalies[i];
        for (var j = 0; j < a.issues.length; j++) {
            var issue = a.issues[j];
            if (issue.targetValue > 0) {
                chartData.push({
                    name: a.campaignName,
                    type: a.bidding.type,
                    avgDaily: a.stats.cost / CONFIG.LOOKBACK_DAYS,
                    targetValue: issue.targetValue,
                    actualValue: issue.actualValue || 0,
                    gap: issue.gap || 0,
                    severity: issue.severity,
                    noData: issue.type === 'UNZUREICHENDE_DATEN',
                });
                break;
            }
        }
    }
    if (chartData.length === 0) return '';
    return buildChartTable(chartData);
}

function buildChartTable(data) {
    if (!data || data.length === 0) return '';
    var BAR_W = 268, LABEL_W = 172, VAL_W = 112;
    var colMed = '#B08800', colHigh = NR.coral;

    var maxGap = 0.80;
    for (var i = 0; i < data.length; i++) {
        if (data[i].gap > maxGap) maxGap = Math.ceil((data[i].gap + 0.05) * 10) / 10;
    }
    var px20 = Math.round((0.20 / maxGap) * BAR_W);
    var px50 = Math.round((0.50 / maxGap) * BAR_W);

    var html = '<table width="552" cellpadding="0" cellspacing="0" border="0" ' +
        'style="border-collapse:collapse;font-family:Arial,Helvetica,sans-serif;">';

    html += '<tr>';
    html += '<td width="' + LABEL_W + '" bgcolor="' + NR.dark + '"
style="background-color:' + NR.dark + ';" +
        'padding:10px 10px 6px;vertical-align:bottom;">' +
        '<p
style="margin:0;font-size:10px;color:#888888;font-family:Arial,Helvetica,sans-serif;">Kam
pagne</p></td>';
    html += '<td width="' + BAR_W + '" bgcolor="' + NR.dark + '" style="background-color:'
+ NR.dark + ';" +
        'padding:10px 0 0;">' +
        '<table width="' + BAR_W + '" cellpadding="0" cellspacing="0" border="0"><tr>
+

```

```

        '<td width="' + px20 + '"' bgcolor="#2d5a27"
style="background-color:#2d5a27;padding:5px 2px;font-size:8px;' +

'font-weight:700;color:#aaffaa;text-align:center;font-family:Arial,Helvetica,sans-serif;' +
>Auf Ziel</td>' +
        '<td width="' + (px50 - px20) + '"' bgcolor="#7a6000"
style="background-color:#7a6000;padding:5px 2px;' +

'font-size:8px;font-weight:700;color:#ffee88;text-align:center;font-family:Arial,Helvetic
a,sans-serif;">Mittel</td>' +
        '<td bgcolor="#7a2820" style="background-color:#7a2820;padding:5px
2px;font-size:8px;font-weight:700;' +
        'color:#ffaaaa;text-align:center;font-family:Arial,Helvetica,sans-serif;">Hohes
Risiko</td>' +
    '</tr></table></td>';
    html += '<td width="' + VAL_W + '"' bgcolor="" + NR.dark + '"' style="background-color:'
+ NR.dark + ';" +
        'padding:10px 10px 6px;text-align:center;vertical-align:bottom;">' +
        '<p
style="margin:0;font-size:10px;color:#888888;font-family:Arial,Helvetica,sans-serif;">Wer
te</p></td>';
    html += '</tr>';

    for (var r = 0; r < data.length; r++) {
        var d = data[r];
        var rowBg = (r % 2 === 0) ? '#ffffff' : '#F8F7F5';
        var gap = d.gap || 0;
        var barPx = d.noData ? 0 : Math.min(Math.round((gap / maxGap) * BAR_W), BAR_W - 2);
        var remPx = BAR_W - barPx;
        var barColor = gap > 0.50 ? colHigh : (gap > 0.20 ? colMed : '#2A7A20');
        var remBg = gap > 0.50 ? '#fdecea' : (gap > 0.20 ? '#fff8e0' : '#e8f5e4');
        var isCpa = d.type === 'TARGET_CPA';
        var tLabel = isCpa ? formatCurrency(d.targetValue) : formatRoas(d.targetValue);
        var aLabel = (d.actualValue > 0) ? (isCpa ? formatCurrency(d.actualValue) :
formatRoas(d.actualValue)) : '&mdash;';
        var pctLbl = '+' + formatPercent(gap);
        var name = d.name.length > 22 ? d.name.substring(0, 20) + '...' : d.name;
        var mType = d.type === 'TARGET_CPA' ? 'tCPA' : 'tROAS';
        var budgetStr = CONFIG.CURRENCY_SYMBOL + Math.round(d.avgDaily) + '/Tag &middot; ' +
mType;
        var labelInside = barPx > 44;

        html += '<tr>';
        html += '<td width="' + LABEL_W + '"' bgcolor="" + rowBg + '"'
style="background-color:' + rowBg + ';" +
            'padding:10px;vertical-align:top;border-top:1px solid #eeeeee;">' +
            '<p style="margin:0;font-size:11px;font-weight:700;color:' + NR.dark + ';" +
            'font-family:Arial,Helvetica,sans-serif;">' + escapeHtml(name) + '</p>' +
            '<p style="margin:3px 0 0 0;font-size:9px;color:' + NR.muted + ';" +
            'font-family:Arial,Helvetica,sans-serif;">' + budgetStr + '</p></td>';

        html += '<td width="' + BAR_W + '"' bgcolor="" + rowBg + '"' style="background-color:'
+ rowBg + ';" +
            'padding:14px 0;vertical-align:middle;border-top:1px solid #eeeeee;">';
        if (d.noData) {
            html += '<p style="margin:0;padding:0 6px;font-size:10px;color:' + NR.muted +
';font-style:italic;' +
                'font-family:Arial,Helvetica,sans-serif;">? Zu wenig Daten</p>';
        } else {
            html += '<table width="' + BAR_W + '"' cellpadding="0" cellspacing="0"
border="0"><tr>';
            if (barPx > 0) {
                html += '<td width="' + barPx + '"' height="16" bgcolor="" + barColor + '"' +
                'style="background-color:' + barColor + ';"height:16px;">';
                if (labelInside) {
                    html += '<p style="margin:0;padding:0
5px;font-size:9px;color:#ffffff;font-weight:700;text-align:right;' +
                        'line-height:16px;font-family:Arial,Helvetica,sans-serif;">' + pctLbl +
'</p>';
                }
                html += '</td>';
            }
            if (remPx > 0) {
                html += '<td bgcolor="" + remBg + '"' style="background-color:' + remBg +
';height:16px;">';
                if (!labelInside) {

```

[illegible]


```

    }
    configs.push({ accountId: accountId, accountName: name || rawId, emails: emailList
});
}
return configs;
}

// =====
// HILFSFUNKTIONEN
// =====
// [Timing] Zeitmessung (im Google-Ads-Runtime ist new Date() verfügbar)
function nowMs() { return (new Date()).getTime(); }
function secs(from, to) { return ((to - from) / 1000).toFixed(1) + 's'; }

function getDateRange(days) {
    var tz = 'Europe/Berlin';
    try { tz = AdsApp.currentAccount().getTimeZone(); } catch (e) {}
    var end = new Date();
    end.setDate(end.getDate() - 1); // gestern
    var start = new Date(end);
    start.setDate(start.getDate() - (days - 1));
    return {
        startDate: Utilities.formatDate(start, tz, 'yyyyMMdd'), // Anzeige/Legacy
        endDate: Utilities.formatDate(end, tz, 'yyyyMMdd'),
        startGaq1: Utilities.formatDate(start, tz, 'yyyy-MM-dd'), // [v2] GAQL-Format
        endGaq1: Utilities.formatDate(end, tz, 'yyyy-MM-dd'),
    };
}

function formatCurrency(value) {
    if (!value || isNaN(value)) return CONFIG.CURRENCY_SYMBOL + '0,00';
    return CONFIG.CURRENCY_SYMBOL + value.toFixed(2)
        .replace('.', ',')
        .replace(/\B(?=(\d{3})+(?!\d))/g, '.');
}

function formatPercent(value) {
    return (value * 100).toFixed(1) + '%';
}

function formatRoas(value) {
    return (value >= 1)
        ? (value * 100).toFixed(0) + '% (' + value.toFixed(2) + 'x)'
        : value.toFixed(2) + 'x';
}

function formatDateDisplay(dateStr) {
    return dateStr.substr(6, 2) + '.' + dateStr.substr(4, 2) + '.' + dateStr.substr(0, 4);
}

function escapeHtml(str) {
    return String(str)
        .replace(/&/g, '&amp;')
        .replace(/</g, '&lt;')
        .replace(/>/g, '&gt;')
        .replace(/"/g, '&quot;');
}

```