RCS Chat: Zukunft der mobilen Kommunikation im Marketing

Category: Online-Marketing
geschrieben von Tobias Hager | 17. August 2025



RCS Chat: Zukunft der mobilen Kommunikation im Marketing

SMS ist das Faxgerät des Smartphones, WhatsApp die geschlossene Party – und RCS Chat ist der Türsteher mit API-Zugang, Brand-Badge und Conversion-Fetisch. Wer 2025 noch Push tanzt und hofft, dass E-Mail den Checkout rettet, verpasst die Bühne, auf der die nächste Welle one-to-one Performance passiert. In diesem Artikel zerlegen wir RCS Chat technisch, strategisch und

operativ, erklären, warum Carrier, Google und bald auch Apple den Kanal neu definieren, und zeigen, wie du RCS Business Messaging sauber in deinen MarTech-Stack schraubst – inklusive Architektur, KPIs, Consent und einem 90-Tage-Plan, der keine Ausreden mehr übrig lässt.

- RCS Chat ist der GSMA-Standard für Rich Communication Services – ein SMS-Nachfolger mit Rich Cards, Buttons, Carousels, Lesebestätigungen und verifizierten Absendern.
- RCS Business Messaging (RBM) kombiniert Marketing, Service und Transaktionen in einem nativen Messaging-Client – ohne App-Download, mit API-first-Ansatz.
- Mit Apple’s angekündigter RCS-Unterstützung wächst die Reichweite massiv, auch wenn A2P-Business-Funktionen zum Start auf iOS noch Fragezeichen tragen.
- Technische Kernstücke: Universal Profile, Verified Sender, MaaP, RBM Agents, SMS-Fallback, Consent-Management und DSGVO-konforme Datenflüsse.
- Use Cases mit echtem Geldwert: Warenkorbabbruch, OTP/2FA, Terminlogistik, Loyalty, Up- und Cross-Sell, Conversational Commerce und Service-Handover.
- KPIs jenseits der Öffnungsrate: Zustellstatus, Gelesen, Interaktionsraten pro Suggested Action, Revenue per Conversation und Multi-Touch-Attribution.
- Kostenmodelle sind carrier- und landesabhängig, ROI entsteht durch höhere Conversion und geringere Reibung gegenüber SMS und Messenger-Gärten.
- Risiken: Carrier-Fragmentierung, Spam-Filtern, fehlende E2EE im RBM, iOS-Roadmap, und die Pflicht zu sauberem Opt-in statt Cowboy-Messaging.
- Integration via CPaaS ist pragmatisch: Twilio, Infobip, Sinch, MessageBird & Co. stellen RBM-APIs, Templates, Senderprüfung und Orchestrierungs-Tools.
- 90 Tage zum Go-Live: Infrastruktur wählen, Use Case priorisieren, Templates zertifizieren, Consent sichern, A/B testen, iterieren, skalieren.

RCS Chat ist nicht der nächste Hype, sondern die normative Kraft des Standards, der SMS aus der Steinzeit holt und Marketing endlich dorthin bringt, wo Nutzer ohnehin sind: in den nativen Messaging-Client. RCS Chat ist kein Messenger, der dich an seine Launen fesselt, sondern ein interoperables Protokoll, das Geräte, Carrier und Marken über ein gemeinsames Universal Profile zusammenführt. In RCS Chat bekommst du Rich Media, strukturierte Interaktion und verifizierte Identität in einem Kanal, der wie SMS funktioniert, aber wie eine Mini-App performt. Für Marketer heißt das: weniger Reibung, mehr Kontrolle, bessere Messbarkeit. Für Entwickler heißt das: APIs, Webhooks, JSON-Payloads und Carrier-Spezifika, die man einmal sauber abstrahieren muss. Für CFOs heißt es: ein neuer Performance-Hebel jenseits der stotternden Paid-Klicks. Und ja, RCS Chat ist schon heute produktiv – nicht “vielleicht irgendwann”.

Die meisten Marketingleute haben RCS Chat als “SMS mit hübschen Bildern” abgetan, weil es Jahre brauchte, bis die Carrier auf Linie waren. Das ist vorbei, seit Google mit RBM Druck gemacht, Verified Sender etabliert und die UX auf Geräteebene stabilisiert hat. RCS Chat liefert Buttons, Carousels,

Rich Cards und vorgeschlagene Antworten direkt in den Default-Messenger von Android, inklusive Zustell- und Lesestatus. Im Backend sprechen wir über RCS Business Messaging, über Agents statt Accounts, über MaaP-Gateways, die Nachrichten zustellen, und über Policies, die Spam unterbinden sollen. Kurz: RCS Chat ist technisch sauber, regulatorisch ernst und kommerziell erwachsen geworden. Wer jetzt wartet, wartet auf seine Wettbewerber.

Und weil jede Buzzword-Party einen Realitätscheck braucht: RCS Chat ist nicht magisch. Es ist ein Kanal mit Regeln, Zertifizierungen, Carrier-spezifischen Gebühren, fehlender Ende-zu-Ende-Verschlüsselung im Business-Kontext und einer iOS-Roadmap, die noch nicht alle A2P-Features garantiert. Aber die Kombination aus nativer Verbreitung, Interaktions-UI und Verifizierung macht RCS Chat für Marketing und CRM so attraktiv, dass die ROI-Rechnung selten gegen dich läuft. Die Conversion von Warenkorbabbrüchen steigt, wenn der Checkout als Button in der Nachricht liegt. Die Supportlast sinkt, wenn vorgeschlagene Antworten statt Freitext genutzt werden. Und die Marke gewinnt, wenn ein offizieller, verifizierter Absender mit Logo statt kryptischer Shortcode im Posteingang erscheint. Willkommen in der Zukunft der mobilen Kommunikation.

RCS Chat im Marketing: Definition, Unterschiede zu SMS und WhatsApp

RCS Chat steht für Rich Communication Services und ist der modernisierte Nachfolger der SMS mit Fokus auf Interaktion, Medien und Identität. Im Marketing-Kontext sprechen wir über RCS Business Messaging, kurz RBM, das A2P-Szenarien für Marken ermöglicht. Anders als SMS bietet RCS Chat Rich Cards mit Bildern, Titeln und Beschreibungen, Carousels für Produkt-Scrollys, Suggested Actions als Buttons und Suggested Replies als Chips für schnelle Antworten. Zustellstatus, Lesebestätigungen, Markensignale wie verifiziertes Häkchen, Logo, Markenname und Farben gehören genauso dazu wie sichere Absender-Authentifizierung. Funktionen, die bei WhatsApp zwar existieren, hängen dort aber am Gängelband eines proprietären Ökosystems und riskanter Blockings. RCS Chat kombiniert hingegen Carrier-Infrastruktur, GSMA-Standards und ein offenes Ökosystem, das du über CPaaS-Provider orchesterst.

Der größte Unterschied zwischen RCS Chat und SMS liegt in der UX-Schicht, die Marketing endlich kontrollierbar macht. Statt langer, kryptischer URLs arbeitest du mit Buttons, Deep Links, Dial-Actions, Maps-Intents oder Kalender-Intents, die nativ in Android greifen. Inhalte sind nicht mehr auf 160 Zeichen beschränkt, und Medien laufen ohne MMS-Fallstricke. Dabei bleibt RCS Chat im Standard-Messenger verankert, was die Einstiegshürde auf null reduziert, weil niemand eine App installieren muss. Gleichzeitig ermöglicht die Verified-Sender-Mechanik, dass Nutzer Vertrauen zu deiner Identität aufbauen, bevor sie überhaupt interagieren. Für Kampagnen bedeutet das bessere CTRs, weniger Friktion und mehr abgeschlossene Transaktionen, vor

allem in Commerce- und Service-Prozessen.

Im Vergleich zu WhatsApp Business API punktet RCS Chat mit weniger Plattformpolitik, aber dafür mit Carrier-Policies und landesspezifischen Besonderheiten. Es gibt keine “24-Stunden-Session-Regel” wie beim grünen Messenger, aber es gibt Template- und Content-Review durch Google beziehungsweise den Carrier, je nach Markt. Die Fallback-Mechanik auf SMS stellt Sicherheiten für nicht RCS-fähige Geräte, erfordert aber sauberes Routing und Failover-Logik im Backend. Zudem sind Business-Nachrichten bei RCS nicht Ende-zu-Ende-verschlüsselt, was für regulierte Branchen Feinsteuerung bei Daten und Inhalte braucht. Kurz gesagt: RCS Chat ist offener, näher am Netz, aber du brauchst ein gutes technisches Fundament, damit die Vorteile zur Geltung kommen.

Technische Architektur von RCS Business Messaging: Universal Profile, MaaP und Agents

Das Herz der Technik ist das GSMA Universal Profile, das festlegt, welche RCS-Funktionen Geräte, Netzbetreiber und Server unterstützen müssen. Dieses Profil regelt Grundfunktionen wie Capability Exchange, Messaging-Formate, Rich Cards, File Transfer und die Zustandsmaschinen für Send, Delivery, Read und Event-Callbacks. Für Unternehmen kommt RCS Business Messaging ins Spiel, das über MaaP (Messaging as a Platform) Gateways läuft. Diese Gateways werden von Carriern, Google oder CPaaS-Anbietern betrieben und stellen REST-APIs bereit, über die deine Anwendung Nachrichten sendet, Konversationen verwaltet und Status-Events empfängt. Die eigentlichen Marketing- und Service-Flows laufen in deinem System oder im Orchestration-Layer des CPaaS, während die Carrier-Infrastruktur die Zustellung in den nativen Client übernimmt.

RBM arbeitet mit dem Konzept der Agents, also registrierten Business-Identitäten, die du je Marke, Land oder Use Case anlegst. Ein Agent wird verifiziert, bekommt Name, Logo, Farben, Kurzbeschreibung und einen Satz berechtigter Kategorien, etwa E-Commerce, Finanzen, Reisen oder Versorger. Vor dem Go-Live erfolgt eine Prüfung, damit Spam und Phishing nicht Fuß fassen. Inhaltlich setzt RBM auf JSON-basierte Nachrichtentypen, darunter stand alone Rich Cards, Media Messages, Carousels und Suggested Actions wie OpenURL, DialPhone, ShareLocation oder CreateCalendarEvent. Jede Nachricht kann serverseitig mit Metadata versehen werden, um Attributionspfade, Kampagnen-IDs und Experiment-Tags zu transportieren, die du später in Analytics zusammenführst. Die Events für “SENT”, “DELIVERED”, “READ” und “INTERACTED” laufen als Webhooks zu dir zurück und sind Gold für Performance-Teams.

Auf Integrationsseite spielt der CPaaS-Markt die erste Geige, weil niemand Lust hat, Carrier-spezifische Besonderheiten selbst zu implementieren. Anbieter wie Twilio, Infobip, Sinch, MessageBird, CM.com oder Gupshup kapseln die Komplexität und bieten neben API-Zugängen auch Template-Management,

Senderverifizierung, Testing-Tools, SMS-Fallback-Routing und Rich-UI-Composer. Viele bringen einen Low-Code-Flowbuilder mit, um Conversational Journeys zu modellieren, inklusive NLP-Connectors, Webhooks und CRM-Adapter. Für anspruchsvolle Setups empfiehlt sich eine eigenständige Orchestrierungsschicht, die Kanäle wie RCS Chat, SMS, WhatsApp, E-Mail und Push zentral koordiniert. Hier laufen auch Consent- und Frequency-Capping-Logiken zusammen, damit du nicht schießt wie mit der Schrotflinte und deine Liste verbrennst. Saubere Architektur verhindert Chaos – und Chaos kostet immer Conversion.

Reichweite, iOS und Android: Abdeckung, Zustellbarkeit und Verifizierung

Android ist die natürliche Heimat von RCS Chat, weil Google den Standard über Messages for Android massiv gepusht und vorinstalliert hat. In vielen Märkten ist die Coverage hoch genug, dass RCS-First-Strategien Sinn ergeben, sofern SMS-Fallback sauber implementiert ist. Der große Hebel kommt mit Apple, das RCS-Unterstützung angekündigt hat, die auf dem GSMA Universal Profile basiert. Zum Zeitpunkt dieser Analyse ist davon auszugehen, dass P2P-Funktionen zuerst landen, während Business Messaging auf iOS zeitversetzt oder eingeschränkt ausgerollt wird. Trotzdem verändert allein die Aussicht auf plattformübergreifende Interoperabilität die strategische Gleichung, weil sie RCS Chat langfristig aus der “Android-only”-Ecke herauslöst. Für Budgets bedeutet das: jetzt Pilotprojekte starten, Learnings sammeln, Templates optimieren und beim iOS-Schalter skalieren.

Zustellbarkeit im RCS-Universum hängt von drei Schichten ab: Gerät, Carrier und Agent-Reputation. Ein Gerät muss RCS-capable sein, was bei aktuellen Android-Versionen Standard ist, sofern Messages und die Carrier-Settings RCS aktiviert haben. Der Carrier muss RBM unterstützen oder via Partnerschaften routen, was in vielen Regionen über Google Jibe oder CPaaS-Backbones geschieht. Und dein Agent muss verifiziert sein, sauber senden und nicht in Spam-Heuristiken laufen. Das klingt wie E-Mail-Deliverability in 2015, ist aber dank starker Verifikation oft weniger fragil, solange du Consent, Frequency und Content-Qualität beachtest. Failover-Strategien greifen, wenn eine Capability-Abfrage negativ ist oder eine Zustellung fehlschlägt: Dann rutscht die Nachricht als SMS mit vereinfachtem Inhalt durch, idealerweise mit passendem Link, um die Conversion nicht zu verlieren.

Verifizierung ist kein “Nice-to-have”, sondern das, was Vertrauen und Klicks holt. Der Verified Sender zeigt deinen Markennamen, Logo und ein Abzeichen direkt im Chat-Header, plus Markenfarben für UI-Elemente der Konversation. On top gibt es in vielen Märkten Directory-Exposure, also auffindbare Agent-Profile, und ein Opt-in-Flow, der nativ abgewickelt werden kann. In regulierten Branchen kommen zusätzliche Prüfungen hinzu, etwa Dokumente oder Handelsregisterauszüge, damit der Carrier nicht zum Spam-Injektor wird.

Technisch gesehen hinterlegst du dabei Signaturen und Endpoint-Domains, damit Phishing schwerer wird und Nutzer echte Marken erkennen. Wer sich vor der Verifizierung drückt, verspielt die Hauptwirkung von RCS Chat: Vertrauen, das konvertiert.

Use Cases, UX-Patterns und Conversion-Mechaniken in RCS Kampagnen

Der Sweet Spot von RCS Chat sind transaktionale und semi-transaktionale Journeys, in denen der Nutzer wenige, strukturierte Entscheidungen trifft. Warenkorbabbruch ist der Klassiker: Du sendest eine Rich Card mit Produktbild, Preis, den zwei relevantesten Benefits und zwei Buttons – “Checkout fortsetzen” und “Frage stellen”. Der erste Button öffnet einen Deep Link in die App oder den Web-Checkout, der zweite führt in eine kurze Konversation mit vorgeschlagenen Antworten, etwa zu Versand, Rückgabe oder Zahlung. OTP/2FA wird sicherer und nutzerfreundlicher, wenn der Code nicht nur als Text kommt, sondern mit auto-collectfähigen Feldern in der App kombiniert wird. Terminbestätigungen, Umbuchungen und Anfahrtsnavigation lassen sich mit Suggested Actions in einem Tap erledigen, ohne den Nutzer in labyrinthartige Mobile-Sites zu zwingen. Loyalty-Programme profitieren von Wallet- oder QR-Intents direkt aus der Nachricht, was Aktivierungsschmerzen reduziert.

UX-Patterns in RCS Chat lehnen sich an Micro-App-Prinzipien an: wenig Text, starke Visuals, klare Hierarchie, maximal drei primäre Aktionen. Carousels sind für Produktauswahl und Varianten ideal, aber du solltest sie nicht als Karussell der Entscheidungen überdrehen. Suggested Replies sind deine UX-Brechstange gegen Tippfaulheit und Support-Kosten, weil sie Pfade vorgeben und NLP-Unsicherheiten kaschieren. Jede Aktion sollte serverseitig als Event landen, damit du Journey-Analysen machen kannst: welcher Button wird zuerst geklickt, welcher Button verkauft, welcher Button Support generiert. Ein Pattern, das in Tests oft gewinnt, ist der “Choice-then-Info”-Ablauf: erst Entscheidungschips anbieten, dann Details nachliefern, statt den Nutzer mit Textwänden zuzuschütten. Und bitte: nie “Mehr erfahren” als einzigen Button, außer du willst Absprung sammeln.

Conversational Commerce ist mit RCS Chat nicht nur Marketing, sondern Prozessoptimierung. Du kannst Zahlungshinweise per Deep Link anstoßen, Unterstützung über Handover an Agent-Systeme triggern und Statusupdates wie “Paket kommt heute” mit Map-Intent und Live-Tracking kombinieren. Servicebots sollten auf RCS die meiste Arbeit erledigen, aber stets einen Hand-off in Live-Chat oder Call ermöglichen, falls Confidence niedrig ist. A/B-Tests betreffen nicht nur Texte, sondern die gesamte Komponentenstruktur: Card vs. Carousel, zwei vs. drei Buttons, Icons ja oder nein, Preis sichtbar oder erst nach Tap. Die Metriken zeigen schnell, dass jede zusätzliche Denkaufgabe Conversion frisst, also designe Konversationen wie Landingpages mit klaren

Funnels. Wer RCS Chat als hübsche SMS missversteht, verschenkt den Hauptvorteil: native, geführte Interaktion.

- Starte mit einem einzigen, wertschöpfenden Use Case: z. B. Warenkorbabbruch oder Terminbestätigung.
- Definiere 1 Ziel, 1 Primäraction, 1 Fallback-Action – mehr ist am Anfang Lärm.
- Baue eine Rich Card mit Bild, Titel, 2 Bullet-Benefits und 2 Buttons.
- Hinterlege UTM-Parameter und Campaign-IDs auf jedem Deep Link.
- Richte Webhooks für SENT, DELIVERED, READ, INTERACTED ein und logge sauber.
- Teste Varianten mit echten Volumina, nicht mit 200 Nachrichten am Freitagabend.

Integration, APIs, Consent und DSGVO: So baust du RCS Chat sauber ein

Technisch beginnt Integration mit der Wahl eines CPaaS-Partners, der dein Zielgebiet, deine Use Cases und dein Volumen abdeckt. Du registrierst einen RBM Agent, lädst Markenassets, definierst Standardantworten und lässt dich verifizieren. Im Backend richtest du die REST-API und Webhooks ein, mapst Events auf deine CDP, dein CRM und dein Analytics, und baust eine Routing-Logik für RCS-fähige Geräte samt SMS-Fallback. Jede Nachricht ist ein JSON-Objekt mit Struktur und Media-Assets, die du aus einem CDN ausspielst, damit die TTFB nicht alles zerstört. Auf der Orchestrierungsebene definierst du, wann RCS Chat gegenüber E-Mail, Push oder WhatsApp priorisiert wird, damit du nicht dieselbe Botschaft viermal verschickst. Ein dediziertes Frequency Capping pro Nutzer und Kanal ist Pflicht, sonst grillst du deine Opt-ins schneller, als du sie sammelst.

Rechtlich gilt: RCS Chat ist Direktmarketing, also brauchst du informierte Einwilligung nach DSGVO und UWG. Double Opt-in ist sinnvoll, weil es Beweislast dreht und Zustellqualität steigert. Du dokumentierst Zeitpunkt, Quelle, Zweck und Kanal, und du bietest jederzeit ein klares Opt-out, das sofort respektiert wird. Consent wird in deiner CDP als Attribut gepflegt und über alle Kanäle synchronisiert, damit kein Team “aus Versehen” ohne Erlaubnis sendet. Datenminimierung ist kein Buzzword, sondern Versicherung: So wenig personenbezogene Daten wie nötig in der RCS-Payload, sensible Informationen niemals unverschlüsselt verschicken, und keine Gesundheits- oder Finanzdetails in Freitext. RBM ist nicht Ende-zu-Ende-verschlüsselt, also behandle Inhalte so, als würden sie auf Logs sitzen, die einmal geprüft werden.

Security-by-Design heißt auch: signierte Webhooks, IP-Allowlists, Rotationszyklen für API Keys, Rate Limits und saubere Observability. Baue ein Error-Budget und automatische Replays für temporäre Carrier-Fehler ein, weil Netze flappen und keine Kampagne immun ist. Monitoring gehört in die CI/CD-

Pipeline; jede Änderung an Template, Bildgröße, Button-Label oder Linkstruktur kann Deliverability oder Performance verschieben. Außerdem wichtig: Brand-Impersonation vermeiden, indem du Domains, Absender und Landingpages konsistent hältst. Und wenn du in mehreren Ländern aktiv bist, prüfe lokale Telekom-Regeln, die A2P-Messaging unterschiedlich regulieren, inklusive Do-Not-Call-Listen und Ruhezeiten. Wer Compliance als Afterthought behandelt, entdeckt sie später im KPI-Friedhof wieder.

Metriken, Tracking und Kostenmodelle: KPI, ROI und Attribution

RCS Chat gibt dir endlich verlässliche Zustell- und Interaktionssignale, die SMS nie hatte. Die Statuskette von SENT über DELIVERED zu READ macht Öffnungsraten messbar, ohne mit Pixeln zu jonglieren. INTERACTED-Events auf Button-Ebene sind dein CTR-Ersatz, nur präziser, weil kein Web-Redirect nötig ist. Revenue per Conversation wird zur Leitkennzahl, wenn du Deep Links mit UTM-Parametern versiehst und in deinem Analytics sauber auflöschst. Optional kannst du Client-Side Events über App Links und Deferred Deep Links mit Attribution-Partnern wie AppsFlyer, Adjust oder Branch verbinden. Das Schöne: Selbst wenn Cookies sterben, lebt die Konversation weiter, weil sie serverseitig sauber instrumentiert ist. Wichtig ist nur, dass du keine private Datenlawine anhäufst, sondern IDs pseudonymisierst und Purpose-bound speicherst.

Attribution in RCS Chat folgt einem einfachen Prinzip: Die Konversation ist der Touchpoint, der Klick der Intent-Beweis, die Conversion die Bestätigung. Multi-Touch-Modelle machen Sinn, wenn du RCS parallel zu E-Mail, Paid Social und Push fährst, aber verliere dich nicht im Modelltheater. Baue zuerst ein robustes First-Touch- oder Last-Non-Direct-Modell und ergänze es später mit datengetriebenen Ansätzen. Auf Kampagnenebene testest du Sendefenster, Frequenzen und Inhalte, auf Journeyebene die Anzahl und Reihenfolge der Nachrichten. Cohort-Analysen zeigen, wie viel Umsatz pro 1.000 versendeten RCS-Nachrichten entsteht, und liefern die Grundlage für Bid- und Budgetentscheidungen in Paid-Kanälen. Und ja, RCS kann Paid-Search-Kosten senken, wenn du Wiederkauf und Service aus dem Suchschlauch befreist.

Die Kostenmodelle sind carrier- und regionsspezifisch und bewegen sich zwischen per-Message und per-Session, manchmal mit Staffelpreisen oder Sonderkonditionen für verifizierte Volumina. Im Schnitt liegt RCS über SMS, aber unter der Summe aus WhatsApp-Sessiongebühren plus Creative-Aufwand, vor allem wenn du Fallback richtig orchestrierst. Rechne TCO, nicht Sticker-Preis: CPaaS-Fees, Carrier-Kosten, Creative-Produktion, Engineering-Zeit und interne Ops gehören in den Topf. Der ROI entsteht, wenn deine Conversion um wenige Prozentpunkte steigt, Supportkosten sinken oder die Abmelderate gegenüber E-Mail niedriger ist. Ein sauberer Business Case vergleicht nicht nur Kanalkosten, sondern End-to-End-Outcome pro Nutzerweg. Wer nur CPMs

schaut, hat Messaging nicht verstanden.

Schritt-für-Schritt-Plan: So führst du RCS Chat in 90 Tagen ein

Ein RCS-Go-Live scheitert selten an Technik, sondern an Unschärfe. Deshalb beginnst du mit einem klaren Ziel, einer eng definierten Journey und einer technischen Basis, die nicht bei erstem Gegenwind kippt. Wähle einen CPaaS-Partner nach Reichweite, SLAs, SDKs und Support, nicht nach der hübschesten Preisfolie. Registriere einen RBM Agent pro Marke und Markt, liefere Markenassets, definiere Standardantworten und baue dein erstes Template als Rich Card mit zwei Buttons. Parallel stellst du Datenflüsse auf, damit Events vom CPaaS in deine CDP, dein CRM und dein BI laufen. Konsent ist Pflicht vor dem ersten Byte, also baue Opt-ins dort, wo der Nutzer echten Wert bekommt: Checkout, Account, Service.

Technisch setzt du auf drei Streams: Messaging, Tracking und Failover. Messaging liefert Payloads, die du in einem strukturierten Template-Repository versionierst, inklusive Render-Tests für verschiedene Geräte. Tracking sendet Events via Webhooks in ein Event-Streaming-System wie Kafka oder direkt in deine CDP, und reichert mit Kampagnen-IDs, Journey-IDs und User-IDs an. Failover prüft vor Versand die RCS-Fähigkeit und schwenkt bei Bedarf auf SMS um, wobei du ein abgespecktes Template mit klarer CTA nutzt. Für das Frontend brauchst du keine App – aber wenn du eine hast, setze App Links und Deferred Deep Links, damit Nutzer nahtlos landen. Auf Serverseite achte auf Rate Limits, Retry-Strategien und Idempotenz, damit du bei Carrier-Zickereien nicht doppelt sendest.

Operativ strukturierst du die 90 Tage in drei Phasen: Setup, Pilot, Scale. In Setup regelst du Verträge, Agent-Verifizierung, Templates, APIs, Webhooks, Consent-Mechanik und QA. Im Pilot wählst du ein Volumen, das statistisch belastbar ist, testest Varianten und dokumentierst Learnings. In Scale setzt du Automatisierung, schaltest weitere Journeys frei, baust Frequency Capping aus und ziehst Kanalswitch-Regeln fest, damit RCS, E-Mail und WhatsApp zusammenarbeiten. Jede Woche ein Release, jede Woche ein Test, jede Woche ein Reporting, das Entscheidungen triggert. Wer wartet, bis “alles perfekt” ist, startet nie – und überlässt die Inbox den anderen.

- Tag 1–10: CPaaS wählen, RBM Agent beantragen, Markenassets liefern, Verträge und SLAs klären.
- Tag 11–20: API und Webhooks integrieren, Event-Schema definieren, Consent-Flows implementieren.
- Tag 21–30: Erstes Template bauen, QA auf Geräten, Fallback-SMS einrichten, Monitoring aufsetzen.
- Tag 31–45: Pilot senden, A/B-Varianten testen, Journey-Events und Revenue mappen, Fixes deployen.
- Tag 46–60: Zweiten Use Case hinzufügen, Frequency Capping schärfen,

Handover in Service etablieren.
- Tag 61–90: Skalieren, Länderausweitung planen, Kostenmodell optimieren, Attributions-Reporting finalisieren.

Fazit: RCS Chat richtig nutzen, bevor es alle tun

RCS Chat ist der erste mobile Kanal seit Jahren, der nicht nur Reichweite verspricht, sondern Interaktion und Identität technisch sauber liefert. Wer Marketing ernsthaft als Produktdisziplin versteht, baut jetzt Journeys, die Reibung entfernen und Conversion dort heben, wo Nutzer täglich sind. Ja, es gibt Baustellen: iOS-Roadmap, Carrier-Fragmentierung, fehlende Ende-zu-Ende-Verschlüsselung im RBM. Aber die Netto-Gleichung geht auf, sobald du Templates, Consent und Messung im Griff hast. Das ist kein “abwarten und Tee trinken”-Thema, sondern ein “testen, lernen, skalieren”-Playbook.

Die Faustregel ist brutal simpel: Wenn du SMS budgetierst, budgetiere RCS Chat – und lass SMS nur der Fallback sein. Rich Cards, Buttons und Verified Sender sind nicht Deko, sondern Geldmaschinen, wenn du sie wie Micro-Apps designst. Und wenn Apple den Schalter für RCS Business Messaging zieht, willst du bereits die beste RCS-UX in deinem Markt fahren, nicht die erste Stolper-Kampagne. Baue heute, dominiere morgen – und überlasse den Rest denen, die noch auf Push hoffen.