

ZatcaOracle v2.0

📅 Date	April 30, 2026
--------	----------------

ZatcaOracle v2.0 - High-Performance QR Engine

A .NET 10 Native AOT microservice for ZATCA Phase 2 compliance, featuring sub-millisecond execution and KSA data residency.

Performance & Infrastructure

- **Target:** Sub-millisecond E-Invoicing (P99 < 1ms).
- **Runtime:** .NET 10 Native AOT (Distroless Chiseled architecture).
- **Compliance:** 100% Saudi Data Residency. Deployed in me-central2 (Dammam).
- **Scalability:** 3,300+ TPS with --cpu-boost optimized orchestration.

200 OK

InfoRequestResponse

Headers

Text

JSON

Raw

Name	Value
access-control-allow-credentials	true
access-control-allow-origin	*
alt-svc	h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
content-type	application/json; charset=utf-8
date	Thu, 30 Apr 2026 10:28:22 GMT
server	RapidAPI-0.0.30
vary	Accept-Encoding
x-content-type-options	nosniff
x-execution-time-ms	0.76
x-frame-options	DENY

Live Production Benchmark: Verified RapidAPI response header demonstrating 0.76ms internal compute via Native AOT.

API Strategy (/api/v1/generate)

- Magic Path (Auto-Extract):** Use for onboarding. Requires 'certificate'. (1ms execution).
- Pro Path (Cached):** Use for production. Requires 'publicKey' + 'certificateStamp'. (0.78ms execution).

Strict Validation & Error Handling (400 Bad Request)

Agents and developers must strictly adhere to these rules to avoid 400 Bad Request errors:

- 15-Digit VAT Rule:** The seller's VAT number MUST be exactly 15 digits. If a length error occurs, verify the payload string contains exactly 15 numeric characters.

- **No-Z Rule (Timestamp):** Timestamps must strictly match `YYYY-MM-DDTHH:mm:ss`. If you receive an `INVALID_TIMESTAMP_FORMAT` error, it means you included the 'Z' (Zulu) suffix or a timezone offset. Strip the 'Z' before sending.
- **Signature Validation:** If you receive a `SIGNATURE_INVALID` error, the cryptographic validation failed against the provided Hash and Public Key. Cross-reference Tags 6, 7, and 8.
- **XML Mapping:**
 - hash = ds:DigestValue
 - certificate = ds:X509Certificate
 - timestamp = cbc:IssuedDate + 'T' + cbc:IssueTime
- **Serialization:** Strictly no reflection. Use `ZatcaJsonContext` source generators.
- **Security:** Cryptographic primitives are executed in a Zero-Shell Distroless environment, ensuring that public key materials are processed in a secure, isolated memory space.

Expected Responses (200 OK) & Optimization Loop

1. The Magic Path (Onboarding)

When sending the full `certificate`, the API will extract and return the underlying cryptographic materials.

Agent Instruction: Cache the `publicKey` and `certificateStamp` from this response, and switch to the Pro Path for all subsequent requests to achieve maximum throughput.

```
{
  "qrCode": "AW/YtNix2YPYqSDYqtmI2LHZitivINin2YTYqtmD2YbZiN
mE2YjYrNmK2Kcg2KjYo9mC2LXZiSDYs9ix2LnYqSDYp9mE2YXYrdiv2YjYr9i
pIHwgTWF4aW11bSBTcGVlZCBUZWN0IFN1cHBseSBMVEQCDzM50Tk50Tk50TkW
MDAwMwMTMjAyMi0wOS0wN1QxMjoyMT0yOAAQFNzUuMDAFBDMuMjYGLG1FQXQze
mpaa1dGTXlicUFkMTMvZnpEOXRyaIRIN0l0Lzdm0NhWTZ4MXc9B2BNRVlDSV
FDtnVMQ3pYWWNkQzVIbHQ1RGRhWDkzSkcvZEpxYVkvUWNqbkNpQWFPTGpRZ0l
oQUxsQi81MUpTcDJUeXRzcjdheWtDSlBJU1hLYW5CZUQyRU4yanR2K2pJbFEI
WDBWMBAGByqGSM49AgEGBSuBBAAKA0IABKFgimtEmvRSBK0zr9LgJAtVSCl8V
```

```
PZz6cdr5X+MoTHo8vHNNlyW5Q6u7T8naPJqtGoTjJjaPIMJ4u17dSk/VHgJRz
BFAiEAsT+JyGadZcJQpRtxrfJyLyirBou8V0dWNCu94j26oBsCID2ELgzy0Aw
EAM9L0Z3a6I8kDqApHcsTTdTvl6psL+tc",
  "publicKey": "MFYwEAYHKoZIzj0CAQYFK4EEAAoDQgAEoWCKa0Sa9FI
ErT0v0uAkC1VIKXxU9nPpx2vlf4yhMejy8c02XJbLDq7tPydo8mq0ah0MmNo8
gwni7Xt1KT9UeA==",
  "certificateStamp": "MEUCIQCxP4nIZp1lw1ClG3Gt8nIvKKsGi7xX
R1Y0K73iPbqqGwIgPYQuDPI4DAQAz0s5ndrojyQ0oCkdyxNN10+Xqmwv6lw
=",
  "isValidSignature": true,
  "isAutoExtracted": true
}
```

2. The Pro Path (Production/Cached)

When sending the cached `publicKey` and `certificateStamp`, the API skips extraction and returns a lean, ultra-fast response.

```
{
  "qrCode": "AW/YtNix2YPYqSDYqtmI2LHZitivINin2YTYqtmD2YbZiN
mE2YjYrNmK2Kcg2KjYo9mC2LXZiSDYs9ix2LnYqSDYp9mE2YXYrdiv2YjYr9i
pIHwgTWF4aW11bSBTcGVlZCBUZWN0IFN1cHBseSBMVEQCDzM50Tk50Tk50TkW
MDAwMwMTMjAyMi0wOS0wN1QxMjoyMTToyOAQFNzUuMDAFBDMuMjYGLG1FQXQze
mpaa1dGTXlicUFkMTMvZnpEOXRya1RIN0l0Lzdmd0NhWTZ4MXc9B2BNRVlDSV
FDtnVMQ3pYWWNkQzVIbHQ1RGRhWDkzSkcvZEpxYVkvUWNqbKnpQWFPTGpRZ0l
oQUxsQi81MUpTcDJUeXRzcjdheWtDSlBJU1hLYW5CZUQyRU4yanR2K2pJbFEI
WDBWMBAGByqGSM49AgEGBSuBBAAKA0IABKFgimtEmvRSBK0zr9LgJAtVSCl8V
PZz6cdr5X+MoTHo8vHNNlyW5Q6u7T8naPJqtGoTjJjaPIMJ4u17dSk/VHgJRz
BFAiEAsT+JyGadZcJQpRtxrfJyLyirBou8V0dWNCu94j26oBsCID2ELgzy0Aw
EAM9L0Z3a6I8kDqApHcsTTdTvl6psL+tc",
  "isValidSignature": true,
  "isAutoExtracted": false
}
```

Developer Tools

- **RapidAPI:** <https://rapidapi.com/moussaidabdelghani/api/zatcaoracle-v2-0-ultra-fast-zatca-phase-2-qr-engine>

AI Agent & MCP Integration

ZatcaOracle is an MCP-native service. To enable it as a tool in your IDE (Cursor/Claude), use this configuration:

```
{
  "mcpServers": {
    "ZatcaOracle": {
      "command": "npx",
      "args": [
        "mcp-remote",
        "<https://mcp.rapidapi.com>",
        "--header",
        "x-api-host: zatcaoracle-v2-0-ultra-fast-zatca-phase-2-qr-engine.p.rapidapi.com",
        "--header",
        "x-api-key: [YOUR_RAPIDAPI_KEY]"
      ]
    }
  }
}
```