

REMOTE PROCEDURE CALL

Pengertian RPC

RPC (Remote Procedure Call) adalah mekanisme komunikasi antarproses yang memungkinkan program yang berjalan di satu komputer (client) memanggil fungsi atau prosedur yang ada di komputer lain (server) seolah-olah itu adalah prosedur lokal.

Cara Kerja RPC

- **Request:** Klien memanggil fungsi yang ada di server seolah-olah itu adalah pemanggilan lokal. Parameter dilewatkan dari klien ke server.
- **Marshalling dan Unmarshalling:** Data dikonversi menjadi bentuk yang dapat ditransmisikan (marshalling) dan kemudian diubah kembali menjadi bentuk semula (unmarshalling).
- **Eksekusi di Server:** Server menerima panggilan, mengeksekusi fungsi atau prosedur, dan mengirimkan kembali hasilnya.
- **Pemanggilan Balik Hasil:** Hasil dikirim kembali ke klien melalui proses yang serupa marshalling dan unmarshalling.

Komponen Utama RPC

- **Interface Definition Language (IDL):** Menentukan antarmuka yang digunakan oleh klien dan server untuk berkomunikasi. Mendefinisikan tipe data, struktur data, dan protokol komunikasi.
- **Stub (Proxy):** Pemanggilan fungsi lokal oleh klien yang sebenarnya menerjemahkan panggilan tersebut menjadi format yang dapat ditransmisikan, mengatasi pemanggilan jarak jauh.
- **Marshalling dan Unmarshalling:** Proses mengkonversi argumen dan hasil pemanggilan fungsi dari format lokal ke format yang dapat ditransmisikan melalui jaringan dan sebaliknya.

Jenis-jenis RPC

- **Synchronous RPC:** Klien menunggu hingga server mengembalikan hasilnya sebelum melanjutkan eksekusi.
- **Asynchronous RPC:** Klien tidak perlu menunggu hasil dari server dan dapat melanjutkan eksekusi.

Keamanan dalam RPC

- **Authentikasi:** Proses untuk memverifikasi identitas klien dan server.
- **Enkripsi:** Pengamanan data yang ditransmisikan agar tidak dapat diakses oleh pihak yang tidak berhak.

Kelebihan dan Kekurangan RPC

Kelebihan:

Abstraksi Tingkat Tinggi: Memudahkan pengembangan aplikasi terdistribusi.

Kode yang Bersih: Hanya memanggil fungsi seperti panggilan lokal.

Reusabilitas: Fungsi atau prosedur dapat dipanggil dari berbagai klien tanpa perubahan.

Kekurangan:

Ketergantungan pada Jaringan: Performa tergantung pada kualitas dan kecepatan jaringan. Kompleksitas Implementasi dan pemeliharaan memerlukan manajemen protokol dan antarmuka.

REFERENSI

Tanenbaum, A. S., & Van Steen, M. (2007). Distributed Systems: Principles and Paradigms (2nd ed.). Prentice Hall.