Manifesto Reaktif

Published on September 16 2014. (v2.0)

Organisasi yang bekerja di berbagai macam bidang yang berbeda-beda, masing-masing telah menemukan pola untuk membangun perangkat lunak yang terlihat sama. Sistem ini lebih kuat, lebih tangguh, lebih fleksibel dan lebih bisa memenuhi tuntutan moderen.

Perubahan ini terjadi karena kebutuhan aplikasi telah berubah secara dramatis dalam beberapa tahun terakhir. Hanya beberapa tahun yang lalu aplikasi berscala besar memiliki puluhan server, waktu respon yang lambat, lamanya pemeliharaan data bergiga-giga byte secara offline.Saat ini, aplikasi dijalankan mulai dari perangkat mobile sampai cluster berbasis cloud yang dijalankan oleh ribuan prosesor multi-core. Ekspektasi pengguna, waktu respon dalam milidetik dan 100% uptime. Data diukur dalam petabyte. Kebutuhan perangkat lunak hari ini tidak dapat dipenuhi oleh arsitektur perangkat lunak lama.

Kami percaya bahwa pendekatan yang koheren pada sistem arsitektur sangat dibutuhkan, dan kami percaya bahwa semua aspek yang diperlukan sudah diakui. Kami menginginkan sistem yang Responsif, Tangguh, Elastis dan Berbasi Pesan. Kami menyebutnya, Reactive Systems.

Sistem yang dibangun sebagai Reactive Systems itu lebih fleksibel, loosely-coupled dan terukur. Hal ini membuat sistem lebih mudah untuk dikembangkan dan terbuka terhadap perubahan. Sistem ini secara signifikan lebih toleran pada kegagalan dan ketika kegagalan tidak dapat dihindari, sistem ini dapat mengatasi kegagalan dengan elegan, terhindar dari bencana. Reactive Systems ini sangat responsif, dapat memberikan para pengguna umpan balik yang efektif secara interaktif.

Reactive Systems adalah:

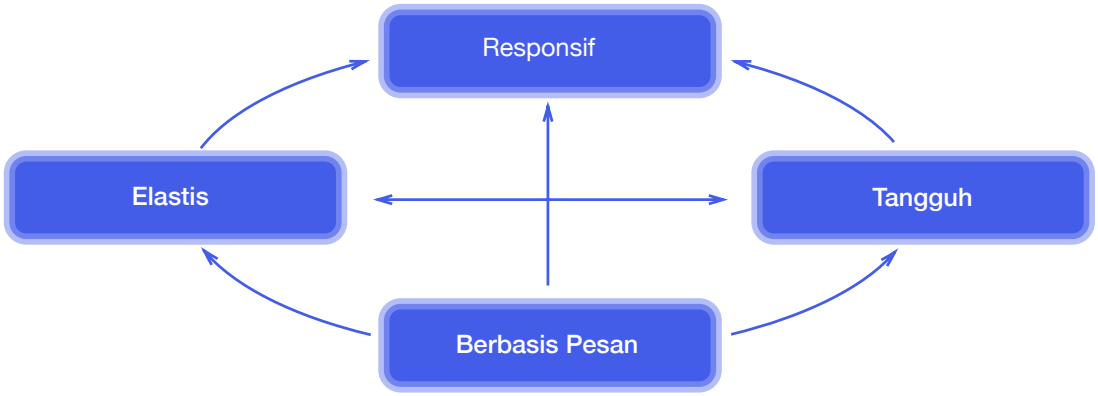
Responsif: Sebuah sistem yang bisa merespon secepat mungkin. Responsiveness adalah landasan kegunaan dan utilitas, tetapi lebih dari itu, respon berarti bahwa masalah dapat dideteksi dengan cepat dan ditangani secara efektif. Sistem responsif fokus pada penyediaan waktu respon yang cepat dan konsisten, membangun sistem yang dapat diandalkan sehingga dapat memberikan kualitas layanan yang konsisten. Kontinuitas perilaku sistem yang seperti ini pada akhirnya akan bisa menyederhanakan penanganan terhadap error, membangun kepercayaan para pengguna, dan mendorong interaksi lebih lanjut.

Tangguh: Sistem ini responsif pada saat menghadapi kegagalan. Hal ini berlaku tidak hanya ketersediaan layanan yang tinggi, seperti mission critical systems — semua sistem yang tidak tangguh akan menjadi tidak responsive pasca kegagalan.

Ketangguhan dapat dicapai dengan cara replikasi, containment, isolasi dan pengalihan. Kegagalan bisa terjadi pada setiap komponent, mengisolasi komponen dari satu sama lain dan dengan demikian memastikan bahwa bagian-bagian dari sistem dapat gagal dan pulih dengan tanpa mengorbankan sistem secara keseluruhan. Pemulihan setiap komponen itu dapat dialihkan kepada komponen lain (eksternal) dan ketersediaan secara maksimal layanan dijamin dengan cara replikasi apabila diperlukan. Para pengguna komponen tersebut tidak terbebani dengan pada saat berlangsungnya penanganan kegagalan pada komponent tersebut.

Elastis: System yang selalu responsif pada situasi yang tidak menentu. Sistem yang reaktif mampu mengatasi perubahan pada tingkat input dengan cara menaikkan atau menurunkan sumber daya yang disediakan untuk melayani berbagai input. Hal ini menggambarkan sebuah desain yang yang tidak memiliki bottle-neck, yang menghasilkan kemampuan untuk mereplikasi atau mengkopi komponen dan mendistribusikan input di antara mereka. Sistem reaktif dapat mensupport prediksi, serta reaktif, mengukur algoritma dengan cara memberikan pengukuran performa sistem yang relevan secara langsung. kemampuan untuk mereplikasi atau mengkopi komponen dan mendistribusikan input di antara mereka. Sistem reaktif dapat mensupport prediksi, serta reaktif, mengukur algoritma dengan cara memberikan pengukuran performa sistem yang relevan secara langsung.

Berbasis Pesan: Sistem reactif bertumpu pada pengiriman-pesan secara asynchronous untuk membangun batas antara komponen-komponent yang memastikan elastisitas, isolasi, transparansi lokasi, dan menyediakan sarana untuk pengalihan kegagalan dalam bentuk pesan. Penerapan pesan-passing secara eksplisit memungkinkan pengaturan beban, elastisitas, dan kontrol aliran dengan membentuk dan memantau antrian pesan dalam sistem dan menerapkan back-pressure bila diperlukan. Lokasi messaging yang transparan sebagai sarana komunikasi memungkinkan bagi para agen penanggulanan kegagalan untuk bekerja dengan konstruksi dan semantik yang sama pada seluruh klaster atau hanya satu klaster. Non-blocking komunikasi memungkinkan penerima bisa menerima resource hanya pada saat aktif, yang mampu mengurangi overhead sistem.



Sistem yang berskala besar terdiri dari komponen-komponen kecil dan karena itu bergantung pada karakter reaktif dari setiap komponent yang dimiliki. Ini berarti bahwa Sistem Reaktif menerapkan prinsip-prinsip desain sehingga sifat ini berlaku di skala semua tingkat, membuat mereka mudah untuk di bangun. Sistem terbesar di dunia bergantung pada arsitektur berdasarkan karakter ini dan melayani kebutuhan milyaran orang setiap hari. Ini adalah saatnya untuk menerapkan prinsip-prinsip desain ini secara sadar mulai dari awal.